



Guia do Desenvolvedor

AWS IoT FleetWise



AWS IoT FleetWise: Guia do Desenvolvedor

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que AWS IoT FleetWise é	1
Benefícios	2
Casos de uso	2
Você é novato na AWS IoT FleetWise?	3
Acessando a AWS IoT FleetWise	3
Preços para AWS IoT FleetWise	3
Principais conceitos	3
Principais conceitos	4
Características da AWS IoT FleetWise	8
Serviços relacionados	9
Configurar a AWS IoT FleetWise	10
Configure sua Conta da AWS	10
Inscreva-se para uma Conta da AWS	10
Criar um usuário com acesso administrativo	11
Comece pelo console	12
Defina suas configurações	12
Definir configurações (console)	13
Definir configurações (AWS CLI)	14
Conceitos básicos	16
Introdução	16
Pré-requisitos	17
Etapa 1: Configurar o software Edge Agent para AWS IoT FleetWise	18
Etapa 2: criar um modelo de veículo	19
Etapa 3: criar um manifesto do decodificador	21
Etapa 4: configurar um manifesto do decodificador	22
Etapa 5: criar um veículo	23
Etapa 6: criar uma campanha	24
Etapa 7: limpar	26
Próximas etapas	26
Ingerir dados	27
Veículos modelo	30
Catálogos de sinais	33
Configurar sinais	36
Crie um catálogo de sinais	42

Importar um catálogo de sinais	47
Atualizar um catálogo de sinais	56
Excluir um catálogo de sinais	58
Obtenha informações do catálogo de sinais	59
Modelos de veículos	60
Criar um modelo de veículo	61
Atualizar o modelo de um veículo	67
Excluir um modelo de veículo	68
Obter informações sobre modelos	69
Manifestos de decodificadores	70
Configurar interfaces e sinais	72
Criar um manifesto do decodificador	75
Atualizar um manifesto do decodificador	83
Excluir um manifesto do decodificador	84
Obtenha informações do manifesto do decodificador	85
Gerenciar veículos	87
Provisionar veículos	88
Autenticar veículos	89
Autorizar veículos	91
Tópicos reservados	92
Criar um veículo	94
Criar um veículo (console)	94
Criar um veículo (AWS CLI)	96
Crie vários veículos	98
Atualizar um veículo	99
Atualize vários veículos	101
Excluir um veículo	102
Excluir um veículo (console)	102
Excluir um veículo (AWS CLI)	102
Obtenha informações sobre o veículo	103
Gerenciar frotas	104
Criar uma frota	105
Associar um veículo a uma frota	106
Desassociar um veículo de uma frota	106
Atualizar uma frota	107
Excluir uma frota	107

Obtenha informações sobre a frota	107
Gerencie dados com campanhas	109
Criar uma campanha	114
Criar uma campanha (console)	115
Criar uma campanha (AWS CLI)	122
Expressões lógicas para campanhas AWS de IoT FleetWise	126
Atualizar uma campanha	127
Excluir uma campanha	128
Excluir uma campanha (console)	128
Excluir uma campanha (AWS CLI)	128
Obtenha informações sobre a campanha	128
Visualize os dados do veículo	130
Processar dados do veículo no Timestream	130
Visualize os dados do veículo armazenados no Timestream	131
Processar dados do veículo no Amazon S3	131
Formato de objeto Amazon S3	132
Analise os dados do veículo armazenados no Amazon S3	132
AWS CLI e SDKs	136
Solução de problemas	137
Problemas com o manifesto do decodificador	137
Problemas com o Edge Agent	141
Problema: o software Edge Agent não inicia.	141
Problema: [ERROR] [IoT FleetWise Engine: :connect]: [Falha ao iniciar a biblioteca de persistência]	143
Problema: O software Edge Agent não coleta diagnósticos integrados (OBD) II PIDs e códigos de problemas de diagnóstico (DTCs).	143
Problema: o FleetWise software Edge Agent para AWS IoT não coleta dados da rede ou não consegue aplicar regras de inspeção de dados.	143
Problema: [ERROR] [AwsIotConnectivityModule: :connect]: [Falha na conexão com erro] ou [WARN] [AwsIotChannel: :send]: [Nenhuma MQTT conexão ativa.]	144
Segurança	145
Proteção de dados	146
Criptografia em repouso na AWS IoT FleetWise	147
Criptografia em trânsito	147
Criptografia de dados na AWS IoT FleetWise	148
Controlar o acesso	156

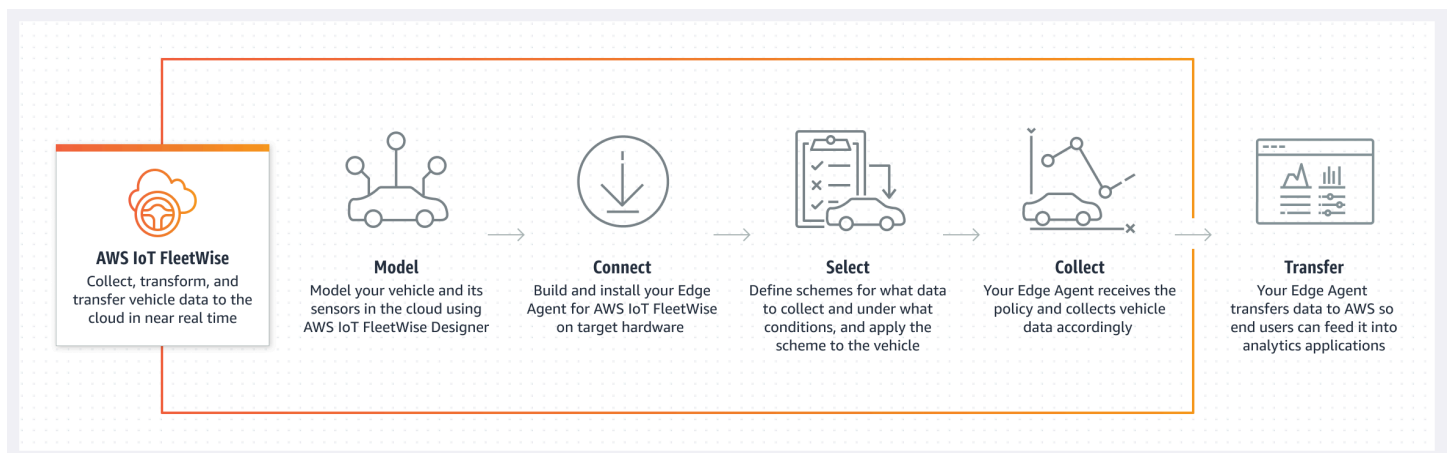
Conceder AWS IoT FleetWise acesso a um destino do Amazon S3	156
Conceder AWS IoT FleetWise acesso a um destino do Amazon Timestream	159
Identity and Access Management	162
Público	163
Autenticando com identidades	163
Gerenciando acesso usando políticas	167
Como a AWS IoT FleetWise funciona com IAM	170
Exemplos de políticas baseadas em identidade	179
Solução de problemas	182
Validação de conformidade	184
Resiliência	186
Segurança da infraestrutura	186
Conectando-se à AWS IoT FleetWise por meio de um endpoint de interface VPC	187
Análise de configuração e vulnerabilidade	190
Melhores práticas de segurança	191
Conceder o mínimo possível de permissões	191
Não registrar em log informações confidenciais	191
Use AWS CloudTrail para ver o histórico API de chamadas	191
Manter o relógio do dispositivo sincronizado	192
Monitorando a AWS IoT FleetWise	193
Monitoramento com CloudWatch	193
Monitore com CloudWatch registros	197
Exibir FleetWise registros AWS de IoT no console CloudWatch	197
Configurar o registro em log do	203
CloudTrail troncos	206
AWS Informações de IoT em FleetWise CloudTrail	206
Entenda as entradas do arquivo de log	207
Histórico do documento	209
.....	ccxi

O que é AWS IoT FleetWise?

AWS IoT FleetWise é um serviço gerenciado que você pode usar para coletar dados do veículo e organizá-los na nuvem. É possível usar os dados coletados para melhorar a qualidade, o desempenho e a autonomia do veículo. Com a AWS IoT FleetWise, você pode coletar e organizar dados de veículos que usam protocolos e formatos de dados diferentes. AWS IoT FleetWise ajuda a transformar mensagens de baixo nível em valores legíveis por humanos e a padronizar o formato de dados na nuvem para análises de dados. Você também pode definir campanhas de coleta de dados para controlar quais dados do veículo serão coletados e quando transferir esses dados para a nuvem.

Quando os dados do veículo estão na nuvem, você pode usá-los para aplicativos que analisam a saúde da frota de veículos. Esses dados ajudam a identificar possíveis problemas de manutenção, tornar mais inteligentes os sistemas de infotenimento em veículos e melhorar tecnologias avançadas, como direção autônoma e sistemas de assistência ao motorista com análises e machine learning (ML).

O diagrama a seguir mostra a arquitetura básica da AWS IoT FleetWise.



Tópicos

- [Benefícios](#)
- [Casos de uso](#)
- [Você é novato na AWS IoT FleetWise?](#)
- [Acessando a AWS IoT FleetWise](#)
- [Preços para AWS IoT FleetWise](#)

- [Principais conceitos e recursos da AWS IoT FleetWise](#)
- [Serviços relacionados](#)

Benefícios

Os principais benefícios da AWS IoT FleetWise são:

Coletar dados do veículo de forma mais inteligente

Melhore a relevância dos dados com a coleta inteligente de dados que envia à nuvem para análise somente os dados de que você precisa.

Analisar facilmente os dados padronizados de toda a frota

Analise dados padronizados de uma frota de veículos sem precisar desenvolver um sistema personalizado de coleta ou registro em log de dados.

Sincronização automática de dados na nuvem

Obtenha uma visão unificada dos dados coletados dos sensores padrão (dados de telemetria) e dos sistemas de visão (dados de câmeras, radares e lidars) e mantenha-os automaticamente sincronizados na nuvem. AWS IoT FleetWise mantém os dados estruturados e não estruturados do sistema de visão, metadados e dados de sensores padrão sincronizados automaticamente na nuvem. Isso simplifica o processo de montar uma visão geral dos eventos e receber informações.

Note

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Casos de uso

Os cenários nos quais você pode usar a AWS IoT FleetWise incluem o seguinte:

Treinar modelos de IA/ML

Melhore continuamente os modelos de machine learning usados em sistemas autônomos e avançados de assistência ao motorista coletando dados de veículos de produção.

Melhorar a experiência digital do cliente

Use dados de sistemas de infotenimento para tornar mais relevantes as informações no aplicativo e os conteúdos audiovisuais no veículo.

Manter a integridade da frota de veículos

Use informações dos dados da frota para monitorar a integridade da bateria de EV e os níveis de carga, gerenciar cronogramas de manutenção, analisar o consumo de combustível e muito mais.

Você é novato na AWS IoT FleetWise?

Se você é iniciante na AWS IoT FleetWise, recomendamos que comece lendo as seguintes seções:

- [Principais conceitos e recursos da AWS IoT FleetWise](#)
- [Configurar a AWS IoT FleetWise](#)
- [Tutorial: Comece a usar a AWS IoT FleetWise](#)
- [Ingira dados AWS de FleetWise IoT na nuvem](#)

Acessando a AWS IoT FleetWise

Você pode usar o FleetWise console de AWS IoT ou acessar API a AWS IoT. FleetWise

Preços para AWS IoT FleetWise

Os veículos enviam dados para a nuvem por meio MQTT de mensagens. Você paga no final de cada mês pelos veículos que criou na AWS IoT FleetWise. Você também paga pelas mensagens coletadas dos veículos. Para obter informações atuais sobre preços, consulte a página de [FleetWise preços de AWS IoT](#). Para saber mais sobre o protocolo MQTT de mensagens, consulte [MQTT](#)o Guia do AWS IoT Core desenvolvedor.

Principais conceitos e recursos da AWS IoT FleetWise

As seções a seguir fornecem uma visão geral dos componentes do FleetWise serviço de AWS IoT e de como eles interagem.

Depois de ler essa introdução, consulte a [Configurar a AWS IoT FleetWise](#) seção para saber como configurar a AWS FleetWise IoT.

Tópicos

- [Principais conceitos](#)
- [Características da AWS IoT FleetWise](#)

Principais conceitos

AWS IoT FleetWise fornece uma estrutura de modelagem de veículos para você modelar seu veículo e seus sensores e atuadores na nuvem. Para permitir a comunicação segura entre seu veículo e a nuvem, a AWS IoT FleetWise também fornece uma implementação de referência para ajudá-lo a desenvolver o software Edge Agent que você pode instalar em seu veículo. Você pode definir esquemas de coleta de dados na nuvem e implantá-los no veículo. O software Edge Agent executado no veículo usa esquemas de coleta de dados para controlar quais dados coletar e quando transferi-los para a nuvem.

A seguir estão os principais conceitos da AWS IoT FleetWise.

Signal

Sinais são estruturas fundamentais que você define para conter dados do veículo e seus metadados. Um sinal pode ser um atributo, uma ramificação, um sensor ou um atuador. Por exemplo, você pode criar um sensor para receber valores de temperatura no veículo e armazenar os metadados, incluindo um nome de sensor, um tipo de dados e uma unidade. Para obter mais informações, consulte [Gerencie AWS catálogos de sinais de IoT FleetWise](#).

Atributo

Os atributos representam informações estáticas que geralmente não mudam, como fabricante e data de fabricação.

Ramificação

As ramificações representam sinais em uma estrutura aninhada. As ramificações demonstram hierarquias de sinais. Por exemplo, a ramificação `Vehicle` tem uma ramificação secundária, `Powertrain`. A ramificação `Powertrain` tem uma ramificação secundária, `combustionEngine`. Para localizar a ramificação `combustionEngine`, use a expressão `Vehicle.Powertrain.combustionEngine`.

Sensor

Os dados do sensor informam o estado atual do veículo e mudam ao longo do tempo, à medida que o estado do veículo muda, como níveis de fluidos, temperaturas, vibrações ou tensão.

Actuator

Os dados do atuador relatam o estado de um dispositivo do veículo, como motores, ar quente e fechaduras de portas. A alteração do estado de um dispositivo do veículo pode atualizar os dados do atuador. Por exemplo, você pode definir um atuador para representar o ar quente. O atuador recebe novos dados quando você liga ou desliga o ar quente.

Estrutura personalizada

Uma estrutura personalizada (também conhecida como struct) representa uma estrutura de dados complexa ou de ordem superior. Ela facilita o agrupamento ou a vinculação lógica de dados originários da mesma fonte. Uma struct é usada quando os dados são lidos ou gravados em uma operação atômica, como para representar um tipo de dados complexo ou uma forma de ordem superior.

Um sinal do tipo struct é definido no catálogo de sinais usando uma referência a um tipo de dados de struct em vez de um tipo de dados primitivo. As structs podem ser usadas para todos os tipos de sinais, incluindo sensores, atributos, atuadores e tipos de dados de sistemas de visão. Se um sinal do tipo estrutura for enviado ou recebido, a AWS FleetWise IoT espera que todos os itens incluídos tenham valores válidos, portanto, todos os itens são obrigatórios. Por exemplo, caso uma struct contenha os itens `Vehicle.Camera.Image.height`, `Vehicle.Camera.Image.width` e `Vehicle.Camera.Image.data`, espera-se que o sinal enviado contenha valores para todos esses itens.

Note

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Propriedade personalizada

Uma propriedade personalizada representa um membro da estrutura de dados complexa. O tipo de dados da propriedade pode ser primitivo ou outra struct.

Ao representar uma forma de ordem superior usando uma struct e uma propriedade personalizada, a forma de ordem superior pretendida é sempre definida e visualizada como uma estrutura em árvore. A propriedade personalizada é usada para definir todos os nós da folha, enquanto a struct é usada para definir todos os nós que não são da folha.

Catálogo de sinais

Um catálogo de sinais contém uma coleção de sinais. Os sinais em um catálogo de sinais podem ser usados para modelar veículos que usam protocolos e formatos de dados diferentes. Por exemplo, há dois carros fabricados por montadoras diferentes: um usa o protocolo Control Area Network (CANbus); o outro usa o protocolo On-board Diagnostics (OBD). Você pode definir um sensor no catálogo de sinais para receber valores de temperatura no veículo. Esse sensor pode ser usado para representar os termopares nos dois carros. Para obter mais informações, consulte [Gerencie AWS catálogos de sinais de IoT FleetWise](#).

Modelo do veículo (manifesto do modelo)

Modelos de veículos são estruturas declarativas que você pode usar para padronizar o formato dos veículos e definir relações entre os sinais nos veículos. Os modelos de veículos impõem informações consistentes em vários veículos do mesmo tipo. Você adiciona sinais para criar modelos de veículos. Para obter mais informações, consulte [Gerencie modelos AWS de veículos de IoT FleetWise](#).

Manifesto do decodificador

Os manifestos do decodificador contêm informações de decodificação para cada sinal nos modelos de veículos. Sensores e atuadores em veículos transmitem mensagens de baixo nível (dados binários). Com os manifestos do decodificador, a AWS IoT FleetWise é capaz de transformar dados binários em valores legíveis por humanos. Cada manifesto do decodificador está associado a um modelo de veículo. Para obter mais informações, consulte [Gerencie manifestos AWS de decodificadores de IoT FleetWise](#).

Interface de rede

Contém informações sobre o protocolo que a rede do veículo usa. AWS IoT FleetWise é compatível com os seguintes protocolos.

Rede de área do controlador (CANbus)

Um protocolo que define como os dados são comunicados entre unidades de controle eletrônico (ECUs). ECUs pode ser a unidade de controle do motor, os airbags ou o sistema de áudio.

Diagnóstico a bordo (I/OBD)

Um protocolo mais desenvolvido que define como os dados de autodiagnóstico são comunicados entre eles. ECUs Ele fornece vários códigos de problemas de diagnóstico padrão (DTCs) que ajudam a identificar o que há de errado com seu veículo.

Middleware de veículo

O middleware de veículo definido como um tipo de interface de rede. Exemplos de middleware de veículos incluem Robot Operating System (ROS2) e Scalable Service-Oriented Middleware over IP (/IP). SOME

Note

AWS A IoT FleetWise oferece suporte a ROS 2 middlewares para dados do sistema de visão.

Sinal do decodificador

Fornece informações detalhadas de decodificação para um sinal específico. Cada sinal especificado no modelo do veículo deve ser emparelhado com um sinal de decodificador. Se o manifesto do decodificador contiver interfaces CAN de rede, ele deverá conter sinais do CAN decodificador. Se o manifesto do decodificador contiver interfaces OBD de rede, ele deverá conter sinais do OBD decodificador.

O manifesto do decodificador deverá conter sinais do decodificador de mensagens se também contiver interfaces de middleware do veículo.

Veículo

Uma representação virtual do veículo físico, como um carro ou caminhão. Veículos são instâncias de modelos de veículos. Veículos criados com base no mesmo modelo de veículo herdam o mesmo grupo de sinais. Cada veículo corresponde a uma coisa da AWS IoT .

Frota

Uma frota representa um grupo de veículos. Antes de poder gerenciar facilmente uma frota de veículos, você deve associar veículos individuais a uma frota.

Campanha

Contém esquemas de coleta de dados. Você define uma campanha na nuvem e a implanta em um veículo ou frota. As campanhas fornecem instruções ao software Edge Agent sobre como selecionar, coletar e transferir dados para a nuvem.

Esquema de coleta de dados

Os esquemas de coleta de dados fornecem instruções ao software Edge Agent sobre como coletar dados. Atualmente, a AWS IoT FleetWise suporta o esquema de coleta baseado em condições e o esquema de coleta baseado em tempo.

Esquema de coleta baseado em condições

Use uma expressão lógica para reconhecer quais dados coletar. O software Edge Agent coleta dados quando a condição é atendida. Por exemplo, se a expressão for `$variable.myVehicle.InVehicleTemperature >35.0`, o software Edge Agent coletará valores de temperatura maiores que 35,0.

Esquema de coleta baseado no tempo

Especifique um período em milissegundos para definir a frequência de coleta de dados. Por exemplo, se o período for de 10.000 milissegundos, o software Edge Agent coletará dados uma vez a cada 10 segundos.

Características da AWS IoT FleetWise

A seguir estão os principais recursos da AWS IoT FleetWise.

Modelagem de veículos

Crie representações virtuais de seus veículos e aplique um formato comum para organizar os sinais dos veículos. AWS IoT é FleetWise compatível com a [Especificação de Sinal do Veículo \(VSS\)](#) que você pode usar para padronizar os sinais do veículo.

Coleta de dados baseada em esquema

Defina esquemas para transferir apenas dados de veículos de alto valor para a nuvem. Você pode definir esquemas baseados em condições para controlar quais dados coletar, como dados de valores de temperatura no veículo superiores a 40 graus. Você também pode definir esquemas baseados em tempo para controlar a frequência de coleta de dados.

Edge Agent para software AWS de IoT FleetWise

O software Edge Agent executado em veículos facilita a comunicação entre veículos e a nuvem. Enquanto os veículos estão conectados à nuvem, o software Edge Agent recebe continuamente esquemas de coleta de dados e os coleta adequadamente.

Serviços relacionados

AWS A IoT se FleetWise integra aos seguintes AWS serviços para melhorar a disponibilidade e a escalabilidade de suas soluções em nuvem.

- **AWS IoT Core**— Registre e controle AWS IoT dispositivos que carregam dados do veículo para a AWS IoT FleetWise. Para obter mais informações, consulte [O que é o AWS IoT?](#) no Guia do desenvolvedor do AWS IoT .
- **Amazon Timestream** — Use um banco de dados de séries temporais para armazenar e analisar os dados do veículo. Para obter mais informações, consulte [O que é o Amazon Timestream?](#) no Guia do desenvolvedor do Amazon Timestream.
- **Amazon S3** — Use um serviço de armazenamento de objetos para armazenar e gerenciar os dados do veículo. Para obter mais informações, consulte [O que é o Amazon S3?](#) no Guia do usuário do Amazon Simple Storage Service.

Configurar a AWS IoT FleetWise

Antes de usar a AWS IoT FleetWise pela primeira vez, conclua as etapas nas seções a seguir.

Tópicos

- [Configure seu Conta da AWS](#)
- [Comece pelo console](#)
- [Defina suas configurações AWS de IoT FleetWise](#)

Configure seu Conta da AWS

Conclua as tarefas a seguir para se inscrever AWS e criar um usuário administrativo.

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a <https://portal.aws.amazon.com/billing/inscrição>.
2. Siga as instruções online.

Parte do procedimento de inscrição envolve receber uma chamada telefônica e inserir um código de verificação no teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário raiz tem acesso a todos os Serviços da AWS e atributos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar [tarefas que exigem acesso de usuário-raiz](#).

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando <https://aws.amazon.com/e> escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login [AWS Management Console](#) como proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira sua senha.

Para obter ajuda ao fazer login usando o usuário raiz, consulte [Fazer login como usuário raiz](#) no Guia do usuário do Início de Sessão da AWS .

2. Ative a autenticação multifator (MFA) para seu usuário root.

Para obter instruções, consulte [Habilitar um MFA dispositivo virtual para seu usuário Conta da AWS root \(console\)](#) no Guia IAM do usuário.

Criar um usuário com acesso administrativo

1. Ative o IAM Identity Center.

Para obter instruções, consulte [Habilitar AWS IAM Identity Center](#) no Guia do usuário do AWS IAM Identity Center .

2. No IAM Identity Center, conceda acesso administrativo a um usuário.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte [Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM](#) no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para entrar com seu usuário do IAM Identity Center, use o login URL que foi enviado ao seu endereço de e-mail quando você criou o usuário do IAM Identity Center.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte [Como fazer login no portal de AWS acesso](#) no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No IAM Identity Center, crie um conjunto de permissões que siga as melhores práticas de aplicação de permissões com privilégios mínimos.

Para obter instruções, consulte [Create a permission set](#) no Guia do usuário do AWS IAM Identity Center .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

Para obter instruções, consulte [Add groups](#) no Guia do usuário do AWS IAM Identity Center .

Note

Você pode usar uma função vinculada a serviços com a IoT AWS . FleetWise As funções vinculadas ao serviço são predefinidas pela AWS IoT FleetWise e incluem as permissões que a IoT AWS precisa para enviar métricas para a Amazon FleetWise . CloudWatch Para obter mais informações, consulte [Usando funções vinculadas a serviços para IoT AWS FleetWise](#).

Comece pelo console

Se você ainda não estiver conectado ao seu Conta da AWS, entre e abra o console de [AWS IoT FleetWise](#) . Para começar a usar a AWS IoT FleetWise, crie um modelo de veículo. Um modelo de veículo padroniza o formato de seus veículos.

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. Em Começar com AWS IoT FleetWise, escolha Começar.

Para obter mais informações sobre a criação de modelo de veículo, consulte [Criar um modelo de veículo \(console\)](#).

Defina suas configurações AWS de IoT FleetWise

Você pode usar o FleetWise console de AWS IoT ou definir configurações API para métricas do Amazon CloudWatch Logs, Amazon CloudWatch Logs e criptografar dados com um. Chave gerenciada pela AWS

Com CloudWatch métricas, você pode monitorar a AWS IoT FleetWise e outros AWS recursos. Você pode usar CloudWatch métricas para coletar e monitorar métricas, como determinar se há um limite de serviço excedido. Para obter mais informações sobre CloudWatch métricas, consulte [Monitore AWS a IoT com a Amazon FleetWise CloudWatch](#).

Com o CloudWatch Logs, AWS a IoT FleetWise envia dados de registro para um grupo de CloudWatch registros, onde você pode usá-los para identificar e mitigar quaisquer problemas. Para obter mais informações sobre CloudWatch registros, consulte [Configurar o AWS registro de IoT FleetWise](#).

Com a criptografia de dados, a AWS IoT FleetWise usa para Chaves gerenciadas pela AWS criptografar dados. Você também pode optar por criar e gerenciar chaves com AWS KMS. Para obter mais informações sobre criptografia, consulte [Criptografia de dados na AWS IoT FleetWise](#).

Definir configurações (console)

Se você ainda não estiver conectado ao seu Conta da AWS, entre e abra o console de [AWS IoT FleetWise](#).

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel esquerdo, selecione Configurações.
3. Em Métricas, escolha Ativar. AWS A IoT associa FleetWise automaticamente uma política CloudWatch gerenciada à função vinculada ao serviço e ativa métricas. CloudWatch
4. Em Registro, selecione Editar.
 - a. Na seção de CloudWatch registro, insira o grupo Registro.
 - b. Selecione Enviar para salvar as alterações.
5. Na seção Criptografia, selecione Editar.
 - a. Selecione o tipo de chave a ser usada. Para obter mais informações, consulte [Gerenciamento de chaves na AWS IoT FleetWise](#).
 - i. Use a AWS chave — a AWS IoT FleetWise possui e gerencia a chave.
 - ii. Escolha uma AWS Key Management Service chave diferente — Você gerencia as AWS KMS keys que estão na sua conta.
 - b. Selecione Enviar para salvar as alterações.

Definir configurações (AWS CLI)

No AWS CLI, registre a conta para definir as configurações.

1. Para definir as configurações, execute o seguinte comando.

```
aws iotfleetwise register-account
```

2. Para verificar suas configurações, execute o comando a seguir para recuperar o status do registro.

Note

A função vinculada ao serviço é usada somente para publicar métricas de AWS FleetWise IoT em CloudWatch. Para obter mais informações, consulte [Usando funções vinculadas a serviços para IoT AWS FleetWise](#).

```
aws iotfleetwise get-register-account-status
```

Example Retorno

```
{
  "accountStatus": "REGISTRATION_SUCCESS",
  "creationTime": "2022-07-28T11:31:22.603000-07:00",
  "customerAccountId": "012345678912",
  "iamRegistrationResponse": {
    "errorMessage": "",
    "registrationStatus": "REGISTRATION_SUCCESS",
    "roleArn": "arn:aws:iam::012345678912:role/AWSIoT FleetwiseServiceRole"
  },
  "lastModificationTime": "2022-07-28T11:31:22.854000-07:00",
}
```

O status de registro pode ser um dos seguintes:

- REGISTRATION_SUCCESS— O AWS recurso foi registrado com sucesso.

- **REGISTRATION_PENDING**— AWS A IoT FleetWise está processando a solicitação de registro. Esse processo leva cerca de 5 minutos para ser concluído.
- **REGISTRATION_FAILURE**— AWS A IoT não FleetWise consegue registrar o AWS recurso. Tente novamente mais tarde.

Tutorial: Comece a usar a AWS IoT FleetWise

Com a AWS IoT FleetWise, você pode coletar, transformar e transferir os dados do seu veículo. Use o tutorial nesta seção para começar a usar a AWS IoT FleetWise.

Consulte os tópicos a seguir para saber mais sobre a AWS IoT: FleetWise

- [Ingira dados AWS de FleetWise IoT na nuvem](#)
- [Modelos de AWS veículos de IoT FleetWise](#)
- [Gerencie AWS veículos de IoT FleetWise](#)
- [Gerencie frotas na IoT AWS FleetWise](#)
- [Colete FleetWise dados de AWS IoT com campanhas](#)

Introdução

Use FleetWise a AWS IoT para coletar, transformar e transferir o formato de dados exclusivo de veículos automatizados para a nuvem quase em tempo real. Você tem acesso a informações de toda a frota. Isso pode ajudar a detectar e amenizar com eficiência os problemas de integridade do veículo, transferir sinais de dados de alto valor e diagnosticar problemas remotamente, tudo isso com redução de custos.

Este tutorial mostra como começar a usar a AWS IoT FleetWise. Você aprenderá como criar um modelo de veículo (manifesto de modelo), um manifesto de decodificador, um veículo e uma campanha.

Para obter mais informações sobre os principais componentes e conceitos da AWS IoT FleetWise, consulte. [Principais conceitos e recursos da AWS IoT FleetWise](#)

Tempo estimado: 45 minutos.

Important

Você será cobrado pelos FleetWise recursos de AWS IoT que essa demonstração cria e consome. Para obter mais informações, consulte [AWS IoT FleetWise](#) na página de preços de AWS FleetWise IoT.

Pré-requisitos

Para concluir este tutorial de conceitos básicos, você precisa:

- Um Conta da AWS. Se você não tiver um Conta da AWS, consulte [Criação de um Conta da AWS](#) no Guia de AWS Account Management referência.
- Acesso a um Região da AWS que ofereça suporte à AWS IoT FleetWise. Atualmente, a AWS IoT FleetWise é suportada no Leste dos EUA (Norte da Virgínia) e na Europa (Frankfurt). Você pode usar o seletor de região no AWS Management Console para alternar para uma dessas regiões. Para obter mais informações, consulte [FleetWise endpoints e cotas de AWS IoT](#).
- Recursos do Amazon Timestream:
 - Um banco de dados do Amazon Timestream. Para obter mais informações, consulte [Criar um banco de dados](#) no Guia do desenvolvedor do Amazon Timestream.
 - Uma tabela do Amazon Timestream criada no Amazon Timestream que armazenará seus dados. Para obter mais informações, consulte [Criar uma tabela](#) no Guia do desenvolvedor do Amazon Timestream.
- Demonstração do software Edge Agent. (As instruções para configurar a demonstração estão na próxima etapa.)
 - Você pode usar a demonstração de início rápido do Explore Edge Agent para explorar a AWS IoT FleetWise e aprender a desenvolver o software Edge Agent para AWS IoT. FleetWise Esta demonstração usa um AWS CloudFormation modelo. Ele orienta você na análise da implementação de referência do Edge Agent, no desenvolvimento do seu Edge Agent e, em seguida, na implantação do software do Edge Agent em um Amazon EC2 Graviton e na geração de dados de amostra do veículo. A demonstração também fornece um script que você pode usar para criar um catálogo de sinais, um modelo de veículo, um manifesto de decodificador, um veículo, uma frota e uma campanha, tudo na nuvem.
 - Para baixar a demonstração, navegue até o console de [AWS IoT. FleetWise](#) Na página inicial do serviço, na FleetWise seção Introdução à AWS IoT, escolha Explore o Edge Agent.

Etapa 1: Configurar o software Edge Agent para AWS IoT FleetWise

Note

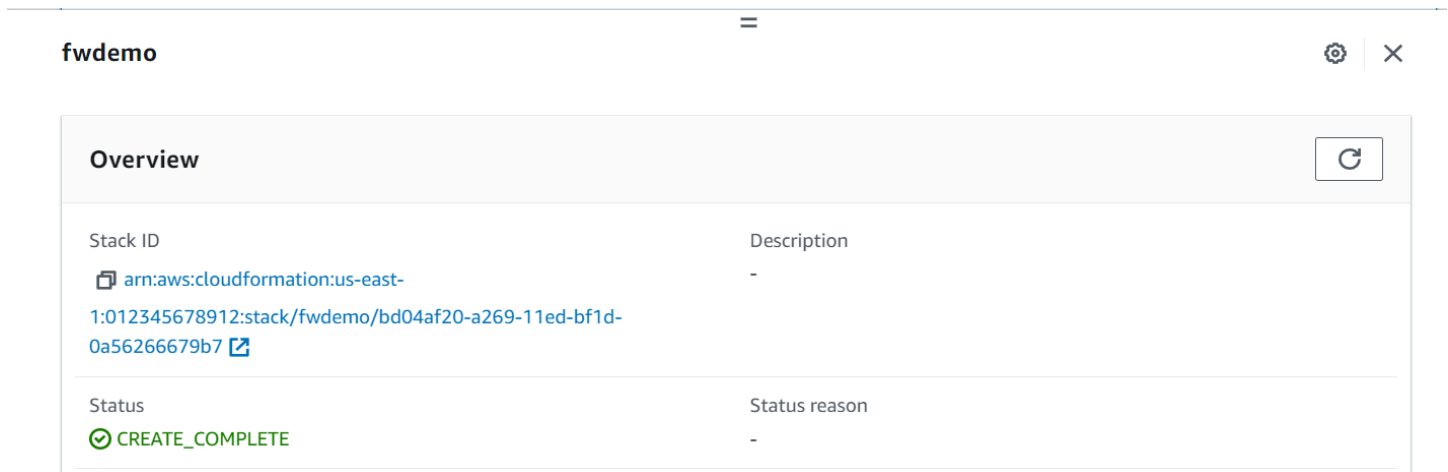
A CloudFormation pilha nesta etapa usa dados de telemetria. Você também pode criar uma CloudFormation pilha usando dados do sistema de visão. Para ter mais informações, consulte o [Vision System Data Developer Guide](#).

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Seu software Edge Agent para AWS IoT FleetWise facilita a comunicação entre veículos e a nuvem. Ele recebe instruções de esquemas de coleta de dados sobre como coletar dados de veículos conectados à nuvem.

Para configurar o software Edge Agent, em Informações gerais, faça o seguinte:

1. Abra o [CloudFormation modelo de lançamento](#).
2. Na página de criação rápida da pilha, em Nome da pilha, insira o nome da sua pilha de recursos de IoT AWS . FleetWise Uma pilha é um nome amigável que aparece como prefixo nos nomes dos recursos que esse AWS CloudFormation modelo cria.
3. Em Parâmetros, insira os valores personalizados para os parâmetros relacionados à pilha.
 - a. Fleetsize: você pode aumentar o número de veículos na frota atualizando o parâmetro Fleetsize.
 - b. oTCoreRegião I - Você pode especificar a região em AWS IoT que a coisa é criada atualizando o parâmetro I oTCore Região. Você deve usar a mesma região que usou para criar seus veículos de AWS IoT FleetWise . Para obter mais informações sobre Regiões da AWS, consulte [Regiões e zonas - Amazon Elastic Compute Cloud](#).
4. Na seção Capacidades, marque a caixa para reconhecer que AWS CloudFormation cria IAM recursos.
5. Escolha Criar pilha e aguarde aproximadamente 15 minutos para que o status da pilha exiba CREATE _ . COMPLETE
6. Para confirmar que a pilha foi criada, escolha a guia Informações da pilha, atualize a visualização e procure por _ . CREATE COMPLETE



The screenshot shows the AWS IoT FleetWise console interface. At the top, there is a header with the text 'fwdemo' on the left and a settings icon and a close icon on the right. Below the header is a section titled 'Overview' with a refresh icon in the top right corner. The 'Overview' section contains two rows of information:

Stack ID	Description
arn:aws:cloudformation:us-east-1:012345678912:stack/fwdemo/bd04af20-a269-11ed-bf1d-0a56266679b7	-
Status	Status reason
✔ CREATE_COMPLETE	-

⚠ Important

Você será cobrado pelos FleetWise recursos de AWS IoT que essa demonstração cria e consome. Para obter mais informações, consulte [AWS IoT FleetWise](#) na página de preços de AWS FleetWise IoT.

Etapa 2: criar um modelo de veículo

⚠ Important


Você não pode criar um modelo de veículo com sinais de dados do sistema de visão no console de AWS IoT FleetWise . Use a AWS CLI em vez disso.

Você usa modelos de veículos para padronizar o formato dos veículos e para ajudar a definir a relação entre os sinais nos veículos que você cria. Um catálogo de sinais também é criado quando você cria um modelo de veículo. Um catálogo de sinais é uma coleção de sinais padronizados que podem ser reutilizados para criar modelos de veículos. Sinais são estruturas fundamentais que você define para conter dados do veículo e seus metadados. No momento, o FleetWise serviço de AWS IoT oferece suporte a apenas um catálogo de sinais Região da AWS por conta. Isso ajuda a verificar se os dados processados de uma frota de veículos são consistentes.

Para criar um modelo de veículo

1. Abra o console AWS de IoT. FleetWise

2. No painel de navegação, selecione Modelos de veículos.
3. Na página Modelos de veículos, selecione Criar modelo de veículo.
4. Na seção Informações gerais, insira o nome do modelo do seu veículo, como Veículo1, e uma descrição opcional. Em seguida, escolha Próximo.
5. Escolha um ou mais sinais do catálogo de sinais. Você pode filtrar sinais por nome no catálogo de pesquisa ou escolhê-los na lista. Por exemplo, você pode escolher sinais de pressão dos pneus e pressão dos freios para poder coletar dados relacionados a esses sinais. Escolha Próximo.
6. Escolha seus arquivos .dbc e carregue-os do seu dispositivo local. Escolha Próximo.

 Note

Para este tutorial, você pode baixar um [exemplo de arquivo.dbc](#) para fazer o upload nesta etapa.

7. Adicione atributos ao modelo do seu veículo e selecione Avançar.
 - a. Nome — Insira o nome do atributo do veículo, como o nome do fabricante ou a data de fabricação.
 - b. Tipo de dados — No menu Tipo de dados, selecione um tipo de dados.
 - c. Unidade — (Opcional) Insira um valor unitário, como quilômetro ou Celsius.
 - d. Caminho — (Opcional) Insira um nome para o caminho para um sinal, como Vehicle.Engine.Light.. O ponto (.) indica que é um sinal secundário.
 - e. Valor padrão — (Opcional) Insira um valor padrão.
 - f. Descrição — (Opcional) Insira uma descrição do atributo.
8. Analise as configurações. Quando estiver pronto, escolha Create (Criar). Uma notificação aparece informando que o modelo do seu veículo foi criado com sucesso.

✔ **Vehicle model created**
You successfully created the vehicle model: demo. ✕

AWS IoT FleetWise > Vehicle models > Demo

demo

[Duplicate](#) [Create vehicle](#) [Create decoder manifest](#)

When a decoder manifest is associated with a vehicle model, you can create a vehicle. To use the API to create vehicles with this vehicle model, follow the instructions in the AWS IoT FleetWise Developer Guide. After you create vehicles, you can create campaigns for them.

Summary [Info](#)

Vehicle model ARN arn:aws:iotfleetwise:us-east-1:012345678912:model-manifest/demo	Status ACTIVE	Date created February 01, 2023 at 14:40 (UTC-05)
Signal catalog ARN arn:aws:iotfleetwise:us-east-1:012345678912:signal-catalog/DefaultSignalCatalog	Description -	Last modified February 01, 2023 at 14:40 (UTC-05)

Etapa 3: criar um manifesto do decodificador

Os manifestos do decodificador estão associados aos modelos de veículos que você cria. Eles contêm informações que ajudam a AWS IoT a FleetWise decodificar e transformar dados de veículos de um formato binário em valores legíveis por humanos que podem ser analisados. As interfaces de rede e sinais de decodificador são componentes que ajudam a configurar manifestos de decodificador. Uma interface de rede contém informações sobre o OBD protocolo CAN ou que a rede do seu veículo usa. O sinal do decodificador fornece informações de decodificação para um sinal específico.

Para criar um manifesto do decodificador

1. Abra o console AWS de IoT. FleetWise
2. No painel de navegação, selecione Modelos de veículos.
3. Na seção Modelos de veículos, selecione o modelo de veículo que você quer usar para criar um manifesto do decodificador.
4. Selecione Criar manifesto do decodificador.

Etapa 4: configurar um manifesto do decodificador

Para configurar um manifesto do decodificador

Important

Você não pode configurar sinais de dados do sistema de visão em manifestos de decodificadores usando o console de IoT AWS . FleetWise Use a AWS CLI em vez disso. Para obter mais informações, consulte [Criar um manifesto do decodificador \(AWS CLI\)](#).

1. Para ajudar a identificar o manifesto do decodificador, insira um nome e uma descrição opcional para ele. Em seguida, escolha Próximo.
2. Para adicionar uma ou mais interfaces de rede, escolha o INTERFACE tipo CAN _ INTERFACE ou OBD _.
 - Interface de diagnóstico a bordo (OBD) - Escolha esse tipo de interface se quiser um protocolo que defina como os dados de autodiagnóstico são comunicados entre as unidades de controle eletrônico (). ECUs Esse protocolo fornece vários códigos de problemas de diagnóstico padrão (DTCs) que podem ajudá-lo a solucionar problemas com seu veículo.
 - Interface de rede de área de controle (CANbus) - Escolha esse tipo de interface se quiser um protocolo que defina como os dados são comunicados entre ECUs eles. ECUs podem ser unidades de controle do motor, airbags ou sistema de áudio.
3. Insira um nome de interface de rede.
4. Para adicionar sinais à interface de rede, escolha um ou mais sinais da lista.
5. Selecione um sinal de decodificador para o sinal adicionado na etapa anterior. Para fornecer informações de decodificação, faça upload de um arquivo .dbc. Cada sinal no modelo do veículo deve ser emparelhado com um sinal de decodificador, que você pode escolher na lista.
6. Para adicionar uma interface de rede secundária, selecione Adicionar interface de rede. Quando terminar de adicionar interfaces de rede, selecione Avançar.
7. Analise as configurações e selecione Criar. É exibida uma notificação informando que o manifesto do decodificador foi criado com sucesso.

Etapa 5: criar um veículo

Na AWS IoT FleetWise, os veículos são representações virtuais do seu veículo físico real. Todos os veículos criados com base no mesmo modelo de veículo herdam o mesmo grupo de sinais, e cada veículo criado corresponde a um item de IoT recém-criado. Você deve associar todos os veículos a um manifesto do decodificador.

Pré-requisitos

1. Verifique se você já criou o modelo do veículo e o manifesto do decodificador. Além disso, verifique se o status do modelo do veículo é ACTIVE.
 - a. Para verificar se o status do modelo do veículo é ACTIVE, abra o console de AWS IoT FleetWise .
 - b. No painel de navegação, selecione Modelos de veículos.
 - c. Na seção Resumo, em Status, verifique o status do seu veículo.

✔ Vehicle model created
You successfully created the vehicle model: demo.

AWS IoT FleetWise > Vehicle models > Demo

demo

[Duplicate](#) [Create vehicle](#) [Create decoder manifest](#)

When a decoder manifest is associated with a vehicle model, you can create a vehicle. To use the API to create vehicles with this vehicle model, follow the instructions in the AWS IoT FleetWise Developer Guide. After you create vehicles, you can create campaigns for them.

Summary Info		
Vehicle model ARN arn:aws:iotfleetwise:us-east-1:012345678912:model-manifest/demo	Status ACTIVE	Date created February 01, 2023 at 14:40 (UTC-05)
Signal catalog ARN arn:aws:iotfleetwise:us-east-1:012345678912:signal-catalog/DefaultSignalCatalog	Description -	Last modified February 01, 2023 at 14:40 (UTC-05)

Para criar um veículo

1. Abra o AWS FleetWise console.
2. No painel de navegação, selecione Veículos.

3. Selecione Criar veículo.
4. Para definir as propriedades do veículo, insira o nome do veículo e selecione um manifesto do modelo (modelo do veículo) e um manifesto do decodificador.
5. (Opcional) Para definir os atributos do veículo, insira um par de valores-chave e selecione Adicionar atributos.
6. (Opcional) Para rotular seu AWS recurso, adicione tags e escolha Adicionar nova tag.
7. Escolha Próximo.
8. Para configurar o certificado do veículo, você pode fazer upload do próprio certificado ou selecionar Gerar automaticamente um novo certificado. Recomendamos gerar automaticamente o certificado para uma configuração mais rápida. Se já tiver um certificado, será possível escolher usá-lo.
9. Faça download dos arquivos de chave pública e privada e selecione Avançar.
10. Para anexar uma política ao certificado do veículo, você pode inserir um nome de política existente ou criar uma nova política. Para criar uma nova política, selecione Criar política > Avançar.
11. Analise as configurações. Após concluir, selecione Criar veículo.

Etapa 6: criar uma campanha

Na AWS IoT FleetWise, as campanhas são usadas para facilitar a seleção, coleta e transferência de dados dos veículos para a nuvem. As campanhas contêm esquemas de coleta de dados que fornecem instruções ao software Edge Agent sobre como coletar dados com um esquema de coleta baseado em condições ou um esquema de coleta baseado em tempo.

Como criar uma campanha

1. Abra o console AWS de IoT. FleetWise
2. No painel de navegação, selecione Campanhas.
3. Escolha Create campaign (Criar campanha).
4. Insira o nome da sua campanha e uma descrição opcional.
5. Para configurar o esquema de coleta de dados da campanha, você pode definir manualmente o esquema de coleta de dados ou fazer upload de um arquivo .json por seu dispositivo local. O upload de um arquivo .json define automaticamente o esquema de coleta de dados.

- a. Para definir manualmente o esquema de coleta de dados, selecione Definir esquema de coleta de dados e selecione o tipo de esquema de coleta de dados que você quer usar para sua campanha. Você pode escolher um esquema de coleta baseado em condições ou um esquema de coleta baseado em tempo.
 - b. Se você escolher um esquema de coleta baseado em tempo, deverá especificar por quanto tempo sua campanha coletará os dados do veículo.
 - c. Se escolher um esquema de coleta baseado em condições, será necessário especificar uma expressão para reconhecer quais dados devem ser coletados. Verifique se o nome do sinal está especificado como uma variável, um operador de comparação e um valor de comparação.
 - d. (Opcional) Escolha a versão do idioma da sua expressão ou mantenha-a como o valor padrão de 1.
 - e. (Opcional) Especifique o intervalo de acionamento entre dois eventos de coleta de dados.
 - f. Para coletar dados, selecione a condição do modo Acionador para o software Edge Agent. Por padrão, o FleetWise software Edge Agent para AWS IoT sempre coleta dados sempre que a condição é atendida. Ou poderá coletar dados somente quando a condição for atendida pela primeira vez, no primeiro acionamento.
 - g. (Opcional) Você pode escolher opções de esquema mais avançadas.
6. Para especificar os sinais dos quais o esquema de coleta de dados coletará dados, procure o nome do sinal no menu.
 7. (Opcional) Você pode escolher uma contagem máxima de amostras ou um intervalo mínimo de amostragem. Você também pode adicionar mais sinais.
 8. Escolha Próximo.
 9. Defina o destino de armazenamento para o qual a campanha deve transferir os dados. Você pode armazenar dados no Amazon S3 ou no Amazon Timestream.
 - a. Amazon S3 — Escolha o bucket do S3 que AWS IoT FleetWise tem permissões para.
 - b. Amazon Timestream — escolha o banco de dados Timestream e o nome da tabela. Insira uma IAM função que permita enviar dados AWS IoT FleetWise para o Timestream.
 10. Escolha Próximo.
 11. Selecione atributos ou nomes de veículos na caixa de pesquisa.
 12. Insira o valor relacionado ao atributo ou nome que você escolheu para seu veículo.
 13. Selecione os veículos dos quais sua campanha coletará dados. Em seguida, escolha Próximo.

14. Analise as configurações da sua campanha e selecione Criar campanha. Você ou sua equipe devem implantar a campanha em veículos.

Etapa 7: limpar

Para evitar cobranças adicionais pelos recursos usados durante este tutorial, exclua a AWS CloudFormation pilha e todos os recursos da pilha.

Para excluir a AWS CloudFormation pilha

1. Abra o [console de AWS CloudFormation](#).
2. Na lista de pilhas, escolha a pilha que você criou na etapa 1.
3. Escolha Excluir.
4. Para confirmar a exclusão, selecione Delete (Excluir). A pilha leva cerca de 15 minutos para ser excluída.

Próximas etapas

1. Você pode processar e visualizar os dados do veículo que sua campanha coleta. Para obter mais informações, consulte [Visualize dados de veículos AWS de FleetWise IoT](#).
2. Você pode solucionar e resolver problemas com a AWS IoT. FleetWise Para obter mais informações, consulte [Solução de problemas de AWS IoT FleetWise](#).

Ingira dados AWS de FleetWise IoT na nuvem

O FleetWise software Edge Agent para AWS IoT, quando instalado e executado em veículos, foi projetado para facilitar a comunicação segura entre seus veículos e a nuvem.

Note

- AWS FleetWise A IoT não se destina ao uso ou em associação com a operação de ambientes perigosos ou sistemas críticos que possam causar lesões corporais graves ou morte ou causar danos ambientais ou materiais. Os dados do veículo coletados por meio do uso da AWS IoT FleetWise são apenas para fins informativos, e você não pode usar a AWS IoT FleetWise para controlar ou operar as funções do veículo.
- Os dados do veículo coletados por meio do uso da AWS IoT FleetWise devem ser avaliados quanto à precisão, conforme apropriado para seu caso de uso, inclusive para fins de cumprimento de quaisquer obrigações de conformidade que você possa ter de acordo com os regulamentos de segurança veicular aplicáveis (como obrigações de monitoramento e emissão de relatórios de segurança). Essa avaliação deve incluir a recolha e análise de informações por meio de outros meios e fontes padrão do setor (tais como relatórios de condutores de veículos).

Para ingerir dados na nuvem, faça o seguinte:

1. Desenvolva e instale seu FleetWise software Edge Agent para AWS IoT em seu veículo. Para obter mais informações sobre como trabalhar com o software Edge Agent, faça o seguinte para baixar o Guia do [desenvolvedor do FleetWise software Edge Agent para AWS IoT](#).
 1. Navegue até o console [AWS de IoT. FleetWise](#)
 2. Na página inicial do serviço, na FleetWise seção Introdução à AWS IoT, escolha Explore o Edge Agent.
2. Crie ou importe um catálogo de sinais contendo sinais que você usará para criar um modelo de veículo. Para ter mais informações, consulte [Crie um catálogo de AWS sinais de IoT FleetWise](#) e [Importar um catálogo de sinais \(AWS CLI\)](#).

Note

- Se você usa o FleetWise console de AWS IoT para criar o primeiro modelo de veículo, não precisa criar manualmente um catálogo de sinais. Quando você cria seu primeiro modelo de veículo, a AWS IoT cria FleetWise automaticamente um catálogo de sinais para você. Para obter mais informações, consulte [Criar um modelo de veículo \(console\)](#).
- AWS FleetWise Atualmente, a IoT suporta um catálogo de sinais para cada AWS conta por. Região da AWS

3. Use sinais no catálogo de sinais para criar um modelo de veículo. Para obter mais informações, consulte [Crie um modelo de AWS veículo de IoT FleetWise](#).

Note

- Se você usar o FleetWise console de AWS IoT para criar um modelo de veículo, poderá carregar arquivos.dbc para importar sinais. .dbc é um formato de arquivo suportado pelos bancos de dados do Controller Area Network (bus). CAN Após a criação do modelo do veículo, novos sinais são adicionados automaticamente ao catálogo de sinais. Para obter mais informações, consulte [Criar um modelo de veículo \(console\)](#).
- Se você usar a CreateModelManifest API operação para criar um modelo de veículo, deverá usar a UpdateModelManifest API operação para ativar o modelo do veículo. Para obter mais informações, consulte [Atualizar um modelo AWS de veículo de IoT FleetWise](#).
- Se você usa o FleetWise console de AWS IoT para criar um modelo de veículo, a AWS IoT ativa FleetWise automaticamente o modelo de veículo para você.

4. Crie um manifesto do decodificador. O manifesto do decodificador contém informações de decodificação para cada sinal especificado no modelo do veículo criado na etapa anterior. O manifesto do decodificador está associado ao modelo de veículo que você criou. Para obter mais informações, consulte [Gerencie manifestos AWS de decodificadores de IoT FleetWise](#).

Note

- Se você usar a CreateDecoderManifest API operação para criar um manifesto do decodificador, deverá usar a UpdateDecoderManifest API operação para ativar

o manifesto do decodificador. Para obter mais informações, consulte [Atualizar um manifesto AWS do decodificador de IoT FleetWise](#).

- Se você usa o FleetWise console de AWS IoT para criar um manifesto do decodificador, a AWS IoT ativa FleetWise automaticamente o manifesto do decodificador para você.

5. Crie veículos com base no modelo do veículo. Veículos criados com base no mesmo modelo de veículo herdam o mesmo grupo de sinais. Você deve usar AWS IoT Core para provisionar seu veículo antes de poder ingerir dados na nuvem. Para obter mais informações, consulte [Gerencie AWS veículos de IoT FleetWise](#).
6. (Opcional) Crie uma frota para representar um grupo de veículos e depois associe veículos individuais à frota. Isso ajuda você a gerenciar vários veículos ao mesmo tempo. Para obter mais informações, consulte [Gerencie frotas na IoT AWS FleetWise](#).
7. Crie campanhas. As campanhas são implantadas em um veículo ou frota de veículos. As campanhas fornecem instruções ao software Edge Agent sobre como selecionar, coletar e transferir dados para a nuvem. Para obter mais informações, consulte [Colete FleetWise dados de AWS IoT com campanhas](#).

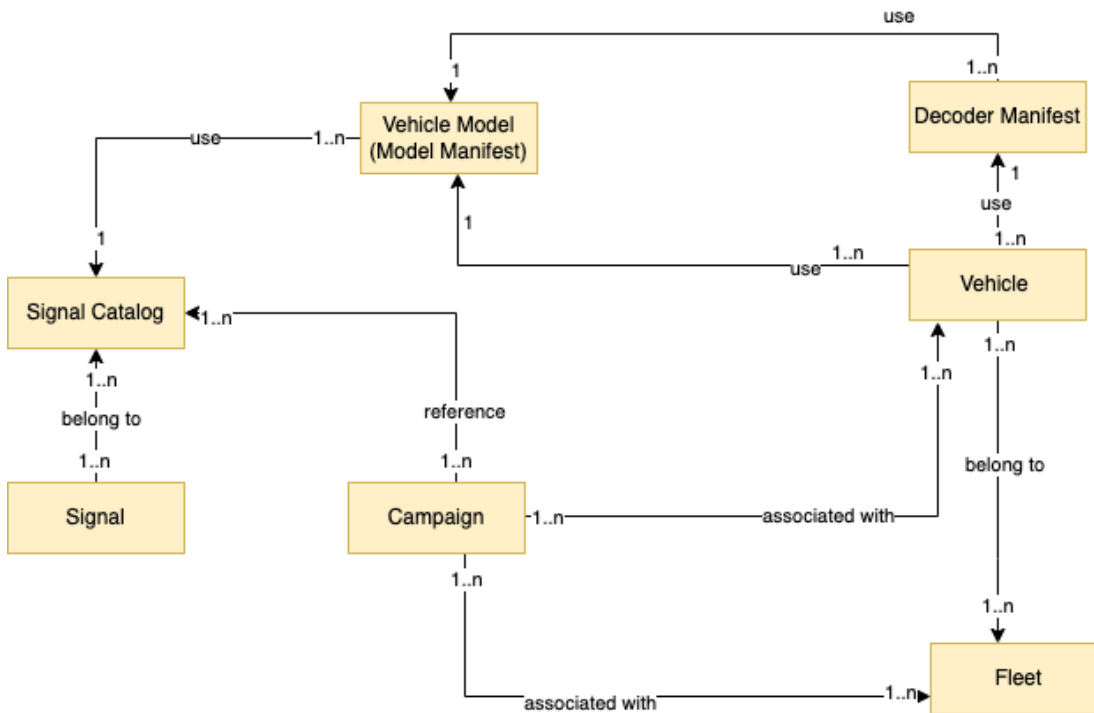
Note

Você deve usar a UpdateCampaign API operação para aprovar a campanha antes que a AWS FleetWise IoT possa implantá-la no veículo ou na frota. Para obter mais informações, consulte [Atualizar uma campanha AWS de IoT FleetWise](#).

O software Edge Agent transfere dados do veículo para AWS IoT Core usar o tópico reservado `aws/iotfleetwise/vehicles/vehicleName/signals`, que envia dados para a AWS IoT FleetWise. Em seguida, a IoT entrega os dados para uma tabela Timestream ou bucket do Amazon S3. Você pode usar o Timestream para consultar seus dados e usar a Amazon ou a QuickSight Grafana para visualizar seus dados. Para obter mais informações, consulte [Visualize dados de veículos AWS de FleetWise IoT](#).

Modelos de AWS veículos de IoT FleetWise

AWS A IoT FleetWise fornece uma estrutura de modelagem de veículos que você pode usar para criar representações virtuais de seus veículos na nuvem. Sinais, catálogos de sinais, modelos de veículos e manifestos de decodificadores são os principais componentes com os quais você trabalha para modelar seus veículos.



Signal

Sinais são estruturas fundamentais que você define para conter dados do veículo e seus metadados. Um sinal pode ser um atributo, uma ramificação, um sensor ou um atuador. Por exemplo, você pode criar um sensor para receber valores de temperatura no veículo e armazenar os metadados, incluindo um nome de sensor, um tipo de dados e uma unidade. Para obter mais informações, consulte [Gerencie AWS catálogos de sinais de IoT FleetWise](#).

Catálogo de sinais

Um catálogo de sinais contém uma coleção de sinais. Os sinais em um catálogo de sinais podem ser usados para modelar veículos que usam protocolos e formatos de dados diferentes. Por exemplo, há dois carros fabricados por montadoras diferentes: um usa o protocolo Control Area Network (CANbus); o outro usa o protocolo On-board Diagnostics (OBD). Você pode definir um sensor no catálogo de sinais para receber valores de temperatura no veículo. Esse sensor pode

ser usado para representar os termopares nos dois carros. Para obter mais informações, consulte [Gerencie AWS catálogos de sinais de IoT FleetWise](#) .

Modelo do veículo (manifesto do modelo)

Modelos de veículos são estruturas declarativas que você pode usar para padronizar o formato dos veículos e definir relações entre os sinais nos veículos. Os modelos de veículos impõem informações consistentes em vários veículos do mesmo tipo. Você adiciona sinais para criar modelos de veículos. Para obter mais informações, consulte [Gerencie modelos AWS de veículos de IoT FleetWise](#) .

Manifesto do decodificador

Os manifestos do decodificador contêm informações de decodificação para cada sinal nos modelos de veículos. Sensores e atuadores em veículos transmitem mensagens de baixo nível (dados binários). Com os manifestos do decodificador, a AWS IoT FleetWise é capaz de transformar dados binários em valores legíveis por humanos. Cada manifesto do decodificador está associado a um modelo de veículo. Para obter mais informações, consulte [Gerencie manifestos AWS de decodificadores de IoT FleetWise](#) .

Você pode usar o FleetWise console de AWS IoT ou API modelar veículos da seguinte maneira.

1. Crie ou importe um catálogo de sinais contendo sinais que você usará para criar um modelo de veículo. Para ter mais informações, consulte [Crie um catálogo de AWS sinais de IoT FleetWise](#) e [Importar um catálogo de sinais \(AWS CLI\)](#).

Note

- Se você usa o FleetWise console de AWS IoT para criar o primeiro modelo de veículo, não precisa criar manualmente um catálogo de sinais. Quando você cria seu primeiro modelo de veículo, a AWS IoT cria FleetWise automaticamente um catálogo de sinais para você. Para obter mais informações, consulte [Criar um modelo de veículo \(console\)](#).
- AWS FleetWise Atualmente, a IoT suporta um catálogo de sinais para cada AWS conta por. Região da AWS

2. Use sinais no catálogo de sinais para criar um modelo de veículo. Para obter mais informações, consulte [Crie um modelo de AWS veículo de IoT FleetWise](#) .

Note

- Se você usar o FleetWise console de AWS IoT para criar um modelo de veículo, poderá carregar arquivos.dbc para importar sinais. .dbc é um formato de arquivo suportado pelos bancos de dados do Controller Area Network (bus). CAN Após a criação do modelo do veículo, novos sinais são adicionados automaticamente ao catálogo de sinais. Para obter mais informações, consulte [Criar um modelo de veículo \(console\)](#).
- Se você usar a CreateModelManifest API operação para criar um modelo de veículo, deverá usar a UpdateModelManifest API operação para ativar o modelo do veículo. Para obter mais informações, consulte [Atualizar um modelo AWS de veículo de IoT FleetWise](#).
- Se você usa o FleetWise console de AWS IoT para criar um modelo de veículo, a AWS IoT ativa FleetWise automaticamente o modelo de veículo para você.

3. Crie um manifesto do decodificador. O manifesto do decodificador contém informações de decodificação para cada sinal especificado no modelo do veículo criado na etapa anterior. O manifesto do decodificador está associado ao modelo de veículo que você criou. Para obter mais informações, consulte [Gerencie manifestos AWS de decodificadores de IoT FleetWise](#).

Note

- Se você usar a CreateDecoderManifest API operação para criar um manifesto do decodificador, deverá usar a UpdateDecoderManifest API operação para ativar o manifesto do decodificador. Para obter mais informações, consulte [Atualizar um manifesto AWS do decodificador de IoT FleetWise](#).
- Se você usa o FleetWise console de AWS IoT para criar um manifesto do decodificador, a AWS IoT ativa FleetWise automaticamente o manifesto do decodificador para você.

CANos bancos de dados de barramento suportam o formato de arquivo.dbc. Você pode fazer upload de arquivos.dbc para importar sinais e sinais de decodificador. Para obter um exemplo de arquivo .dbc, faça o seguinte.

Para obter um arquivo.dbc

1. Faça o download do [EngineSignalsarquivo.zip](#).

2. Navegue até o diretório em que você fez o download do arquivo `EngineSignals.zip`.
3. Descompacte o arquivo e salve-o localmente como `EngineSignals.dbc`.

Tópicos

- [Gerencie AWS catálogos de sinais de IoT FleetWise](#)
- [Gerencie modelos AWS de veículos de IoT FleetWise](#)
- [Gerencie manifestos AWS de decodificadores de IoT FleetWise](#)

Gerencie AWS catálogos de sinais de IoT FleetWise

Note

Você pode baixar um [script de demonstração](#) para converter ROS 2 mensagens em VSS JSON arquivos compatíveis com o catálogo de sinais. Para ter mais informações, consulte o [Vision System Data Developer Guide](#).

Um catálogo de sinais é uma coleção de sinais padronizados que podem ser reutilizados para criar modelos de veículos. AWS A IoT é FleetWise compatível com a [Especificação de Sinal de Veículo \(VSS\)](#) que você pode seguir para definir sinais. Um sinal pode ser de qualquer um dos seguintes tipos.

Atributo

Os atributos representam informações estáticas que geralmente não mudam, como fabricante e data de fabricação.

Ramificação

As ramificações representam sinais em uma estrutura aninhada. As ramificações demonstram hierarquias de sinais. Por exemplo, a ramificação `Vehicle` tem uma ramificação secundária, `Powertrain`. A ramificação `Powertrain` tem uma ramificação secundária, `combustionEngine`. Para localizar a ramificação `combustionEngine`, use a expressão `Vehicle.Powertrain.combustionEngine`.

Sensor

Os dados do sensor informam o estado atual do veículo e mudam ao longo do tempo, à medida que o estado do veículo muda, como níveis de fluidos, temperaturas, vibrações ou tensão.

Actuador

Os dados do atuador relatam o estado de um dispositivo do veículo, como motores, ar quente e fechaduras de portas. A alteração do estado de um dispositivo do veículo pode atualizar os dados do atuador. Por exemplo, você pode definir um atuador para representar o ar quente. O atuador recebe novos dados quando você liga ou desliga o ar quente.

Estrutura personalizada

Uma estrutura personalizada (também conhecida como struct) representa uma estrutura de dados complexa ou de ordem superior. Ela facilita o agrupamento ou a vinculação lógica de dados originários da mesma fonte. Uma struct é usada quando os dados são lidos ou gravados em uma operação atômica, como para representar um tipo de dados complexo ou uma forma de ordem superior.

Um sinal do tipo struct é definido no catálogo de sinais usando uma referência a um tipo de dados de struct em vez de um tipo de dados primitivo. As structs podem ser usadas para todos os tipos de sinais, incluindo sensores, atributos, atuadores e tipos de dados de sistemas de visão. Se um sinal do tipo estrutura for enviado ou recebido, a AWS FleetWise IoT espera que todos os itens incluídos tenham valores válidos, portanto, todos os itens são obrigatórios. Por exemplo, caso uma struct contenha os itens `Vehicle.Camera.Image.height`, `Vehicle.Camera.Image.width` e `Vehicle.Camera.Image.data`, espera-se que o sinal enviado contenha valores para todos esses itens.

Note

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Propriedade personalizada

Uma propriedade personalizada representa um membro da estrutura de dados complexa. O tipo de dados da propriedade pode ser primitivo ou outra struct.

Ao representar uma forma de ordem superior usando uma struct e uma propriedade personalizada, a forma de ordem superior pretendida é sempre definida e visualizada como uma estrutura em árvore. A propriedade personalizada é usada para definir todos os nós da folha, enquanto a struct é usada para definir todos os nós que não são da folha.

Note

- Se você usa o FleetWise console de AWS IoT para criar o primeiro modelo de veículo, não precisa criar manualmente um catálogo de sinais. Quando você cria seu primeiro modelo de veículo, a AWS IoT cria FleetWise automaticamente um catálogo de sinais para você. Para obter mais informações, consulte [Criar um modelo de veículo \(console\)](#).
- Se você usar o FleetWise console de AWS IoT para criar um modelo de veículo, poderá carregar arquivos.dbc para importar sinais. .dbc é um formato de arquivo suportado pelos bancos de dados do Controller Area Network (bus). CAN Após a criação do modelo do veículo, novos sinais são adicionados automaticamente ao catálogo de sinais. Para obter mais informações, consulte [Criar um modelo de veículo \(console\)](#).
- AWS FleetWise Atualmente, a IoT oferece suporte a um catálogo de sinais para Conta da AWS cada região.

AWS A IoT FleetWise fornece as seguintes API operações que você pode usar para criar e gerenciar catálogos de sinais.

- [CreateSignalCatalog](#)— Cria um novo catálogo de sinais.
- [ImportSignalCatalog](#)— Importa sinais para criar um catálogo de sinais fazendo o upload de um JSON arquivo. Os sinais devem ser definidos seguindo VSS e salvos no JSON formato.
- [UpdateSignalCatalog](#)— Atualiza um catálogo de sinais existente atualizando, removendo ou adicionando sinais.
- [DeleteSignalCatalog](#)— Exclui um catálogo de sinais existente.
- [ListSignalCatalogs](#)— Recupera uma lista paginada de resumos de todos os catálogos de sinais.
- [ListSignalCatalogNodes](#)— Recupera uma lista paginada de resumos de todos os sinais (nós) em um determinado catálogo de sinais.
- [GetSignalCatalog](#)— Recupera informações sobre um catálogo de sinais.

Tutoriais

- [Configurar AWS sinais de IoT FleetWise](#)
- [Crie um catálogo de AWS sinais de IoT FleetWise](#)
- [Importar um catálogo AWS de sinais de IoT FleetWise](#)
- [Atualizar um catálogo AWS de sinais de IoT FleetWise](#)
- [Excluir um catálogo AWS de sinais de IoT FleetWise](#)
- [Obtenha informações AWS do catálogo de FleetWise sinais de IoT](#)

Configurar AWS sinais de IoT FleetWise

Esta seção mostra como configurar ramificações, atributos, sensores e atuadores.

Tópicos

- [Configurar ramificações](#)
- [Configurar atributos](#)
- [Configurar sensores ou atuadores](#)
- [Configurar tipos de dados complexos](#)

Configurar ramificações

Para configurar uma nova conexão, especifique as seguintes informações:

- `fullyQualifiedName` — O nome totalmente qualificado da ramificação é o caminho para a ramificação além do nome da ramificação. Use um ponto (.) para se referir a uma ramificação secundária. Por exemplo, `Vehicle.Chassis.SteeringWheel` é o nome totalmente qualificado da ramificação `SteeringWheel`. `Vehicle.Chassis.` é o caminho para essa ramificação.

O nome totalmente qualificado pode ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, dois pontos (:) e sublinhado (_).

- (Opcional) `Description` — Descrição da ramificação.

A descrição pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), _ (sublinhado) e - (hífen).

- (Opcional) `deprecationMessage` — A mensagem de descontinuação do nó ou da ramificação que está sendo movida ou excluída.

Eles deprecationMessage podem ter até 2048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), _ (sublinhado) e - (hífen).

- (Opcional) comment — Um comentário adicional à descrição. Um comentário pode ser usado para fornecer informações adicionais sobre a ramificação, como a justificativa da ramificação ou referências a ramificações relacionadas.

O comentário pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), _ (sublinhado) e - (hífen).

Configurar atributos

Para configurar um atributo, especifique as informações a seguir.

- dataType— O tipo de dados do atributo deve ser um dos seguintes: INT8,UINT8,INT16,UINT16,INT32,UINT32,INT64,UINT64,BOOLEAN,FLOAT, DOUBLESTRING, UNIX_TIMESTAMP, INT8_ARRAY, UINT8_ARRAY, INT16_ARRAY, UINT16_ARRAY, INT32_ARRAY, UINT32_ARRAY, INT64_ARRAY, UINT64_ARRAY, BOOLEAN_ARRAY, FLOAT_ARRAY, DOUBLE_ARRAY, STRING_ARRAY, UNIX_TIMESTAMP_ARRAY, UNKNOWN fullyQualifiedName, ou uma estrutura personalizada definida na ramificação do tipo de dados.
- fullyQualifiedName— O nome totalmente qualificado do atributo é o caminho para o atributo além do nome do atributo. Use um ponto (.) para se referir ao sinal secundário. Por exemplo, Vehicle.Chassis.SteeringWheel.Diameter é o nome totalmente qualificado do atributo Diameter. Vehicle.Chassis.SteeringWheel. é o caminho para esse atributo.

O nome totalmente qualificado pode ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e _ (sublinhado).

- (Opcional) Description — A descrição do atributo.

A descrição pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), _ (sublinhado) e - (hífen).

- (Opcional) unit — A unidade científica do atributo, como km ou Celsius.
- (Opcional) min — O valor mínimo do atributo.
- (Opcional) max — O valor máximo do atributo.
- (Opcional) defaultValue — O valor padrão do atributo.
- (Opcional) assignedValue — O valor atribuído ao atributo.

- (Opcional) `allowedValues` — Uma lista de valores que o atributo aceita.
- (Opcional) `deprecationMessage` — A mensagem de descontinuação do nó ou ramificação que está sendo movido ou excluído.

Eles `deprecationMessage` podem ter até 2048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), _ (sublinhado) e - (hífen).

- (Opcional) `comment` — Um comentário adicional à descrição. Um comentário pode ser usado para fornecer informações adicionais sobre o atributo, como a justificativa para o atributo ou referências a atributos relacionados.

O comentário pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), _ (sublinhado) e - (hífen).

Configurar sensores ou atuadores

Para configurar um sensor ou atuador, especifique as seguintes informações.

- `dataType`— O tipo de dados do sinal deve ser um dos seguintes: `INT8`, `UINT8`, `INT16`, `UINT16`, `INT32`, `UINT32`, `INT64`, `UINT64`, `BOOLEAN`, `FLOATDOUBLE`, `UNIX_STRING`, `_TIMESTAMP`, `INT8_ARRAY`, `UINT8_ARRAY`, `INT16_ARRAY`, `UINT16_ARRAY`, `INT32_ARRAY`, `UINT32_ARRAY`, `INT64_ARRAY`, `UINT64_ARRAY`, `BOOLEAN_ARRAY`, `FLOAT_ARRAY`, `DOUBLE_ARRAY`, `STRING_ARRAY`, `UNIX_TIMESTAMP_ARRAY`, `UNKNOWN fullyQualifiedName`, ou uma estrutura personalizada definida na ramificação do tipo de dados.
- `fullyQualifiedName` — O nome totalmente qualificado do sinal é o caminho para o sinal além do nome do sinal. Use um ponto (.) para se referir ao sinal secundário. Por exemplo, `Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState` é o nome totalmente qualificado do atuador `HandsOffSteeringState`. `Vehicle.Chassis.SteeringWheel.HandsOff` é o caminho para esse atuador.

O nome totalmente qualificado pode ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e _ (sublinhado).

- (Opcional) `Description` — A descrição do sinal.

A descrição pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), _ (sublinhado) e - (hífen).

- (Opcional) `unit` — A unidade científica do sinal, como km ou Celsius.

- (Opcional) `min` — O valor mínimo do sinal.
- (Opcional) `max` — O valor máximo do sinal.
- (Opcional) `assignedValue` — O valor atribuído ao sinal.
- (Opcional) `allowedValues` — lista de valores que o sinal aceita.
- (Opcional) `deprecationMessage` — A mensagem de descontinuação do nó ou ramificação que está sendo movido ou excluído.

Eles `deprecationMessage` podem ter até 2048 caracteres. Os caracteres válidos são `a-z`, `A-Z`, `0-9`, `:` (dois pontos), `_` (sublinhado) e `-` (hífen).

- (Opcional) `comment` — Um comentário adicional à descrição. Um comentário pode ser usado para fornecer informações adicionais sobre o sensor ou atuador, como justificativa ou referências a sensores ou atuadores relacionados.

O comentário pode ter até 2.048 caracteres. Os caracteres válidos são `a-z`, `A-Z`, `0-9`, `:` (dois pontos), `_` (sublinhado) e `-` (hífen).

Configurar tipos de dados complexos

Tipos de dados complexos são usados na modelagem de sistemas de visão. Além das ramificações, esses tipos de dados são compostos de estruturas (também conhecidas como `struct`) e propriedades. Uma `struct` é um sinal descrito por vários valores, como uma imagem. Uma propriedade representa um membro da estrutura, como um tipo de dados primitivo (como `UINT8`) ou outra estrutura (como `timestamp`). Por exemplo, `Vehicle.Cameras.Front` representa uma ramificação, `Vehicle.Cameras.Front.Image` representa uma `struct` e `Vehicle.Cameras.Timestamp` representa uma propriedade.

O exemplo de tipo de dados complexo a seguir demonstra como sinais e tipos de dados são exportados para um único JSON arquivo.

Exemplo tipo de dados complexos

```
{
  "Vehicle": {
    "type": "branch"
    // Signal tree
  },
  "ComplexDataTypes": {
    "VehicleDataTypes": {
```

```
// complex data type tree
"children": {
  "branch": {
    "children": {
      "Struct": {
        "children": {
          "Property": {
            "type": "property",
            "datatype": "Data type",
            "description": "Description",
            //          ...
          }
        },
        "description": "Description",
        "type": "struct"
      }
    }
  },
  "description": "Description",
  "type": "branch"
}
}
}
}
```

Note

Você pode baixar um [script de demonstração](#) para converter ROS 2 mensagens em VSS JSON arquivos compatíveis com o catálogo de sinais. Para ter mais informações, consulte o [Vision System Data Developer Guide](#).

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Configurar struct

Para configurar uma estrutura personalizada (ou struct), especifique as informações a seguir.

- `fullyQualifiedName`: o nome totalmente qualificado da estrutura personalizada. Por exemplo, o nome totalmente qualificado de uma estrutura personalizada pode ser `ComplexDataTypes.VehicleDataTypes.SVMCamera`.

O nome totalmente qualificado pode ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e _ (sublinhado).

- (Opcional) `Description` — A descrição do sinal.

A descrição pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), _ (sublinhado) e - (hífen).

- (Opcional) `deprecationMessage` — A mensagem de descontinuação do nó ou ramificação que está sendo movido ou excluído.

Eles `deprecationMessage` podem ter até 2048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), _ (sublinhado) e - (hífen).

- (Opcional) `comment` — Um comentário adicional à descrição. Um comentário pode ser usado para fornecer informações adicionais sobre o sensor ou atuador, como justificativa ou referências a sensores ou atuadores relacionados.

O comentário pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), _ (sublinhado) e - (hífen).

Configurar propriedade

Para configurar uma propriedade personalizada, especifique as informações a seguir.

- `dataType`— O tipo de dados do sinal deve ser um dos seguintes: `INT8`, `UINT8`, `INT16`, `UINT16`, `INT32`, `UINT32`, `INT64`, `UINT64`, `BOOLEAN`, `FLOATDOUBLE`, `UNIX_STRING`, `_TIMESTAMP`, `INT8_ARRAY`, `UINT8_ARRAY`, `INT16_ARRAY`, `UINT16_ARRAY`, `INT32_ARRAY`, `UINT32_ARRAY`, `INT64_ARRAY`, `UINT64_ARRAY`, `BOOLEAN_ARRAY`, `FLOAT_ARRAY`, `DOUBLE_ARRAY`, `STRING_ARRAY`, `UNIX_TIMESTAMP_ARRAYSTRUCT`, `STRUCT_ARRAY`, ou `UNKNOWN`.
- `fullyQualifiedName`: o nome totalmente qualificado da propriedade personalizada. Por exemplo, o nome totalmente qualificado de uma propriedade personalizada pode ser `ComplexDataTypes.VehicleDataTypes.SVMCamera.FPS`.

O nome totalmente qualificado pode ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e _ (sublinhado)

- (Opcional) `Description` — A descrição do sinal.

A descrição pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), _ (sublinhado) e - (hífen).

- (Opcional) `deprecationMessage` — A mensagem de descontinuação do nó ou ramificação que está sendo movido ou excluído.

Eles `deprecationMessage` podem ter até 2048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), _ (sublinhado) e - (hífen).

- (Opcional) `comment` — Um comentário adicional à descrição. Um comentário pode ser usado para fornecer informações adicionais sobre o sensor ou atuador, como justificativa ou referências a sensores ou atuadores relacionados.

O comentário pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), _ (sublinhado) e - (hífen).

- (Opcional) `dataEncoding`: indica se a propriedade é um dado binário. A codificação de dados da propriedade personalizada deve ser uma das seguintes: `BINARY` ou `TYPED`.
- (Opcional) `structFullyQualifiedName` — O nome totalmente qualificado do nó de estrutura (estrutura) da propriedade personalizada se o tipo de dados da propriedade personalizada for `Estrutura` ou `StructArray`

O nome totalmente qualificado pode ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e _ (sublinhado).

Crie um catálogo de AWS sinais de IoT FleetWise

Você pode usar a [CreateSignalCatalog](#) API operação para criar um catálogo de sinais. O exemplo a seguir usa AWS CLI.

Para criar um catálogo de sinais, execute o comando a seguir.

Substituir *signal-catalog-configuration* com o nome do JSON arquivo que contém a configuração.

```
aws iotfleetwise create-signal-catalog --cli-input-json file://signal-catalog-configuration.json
```

- Substituir *signal-catalog-name* com o nome do catálogo de sinais que você está criando.

- (Opcional) Substituir *description* com uma descrição para ajudá-lo a identificar o catálogo de sinais.

Para obter mais informações sobre como configurar ramificações, atributos, sensores e atuadores, consulte [Configurar AWS sinais de IoT FleetWise](#).

```
{
  "name": "signal-catalog-name",
  "description": "description",
  "nodes": [
    {
      "branch": {
        "fullyQualifiedName": "Types"
      }
    },
    {
      "struct": {
        "fullyQualifiedName": "Types.sensor_msgs_msg_CompressedImage"
      }
    },
    {
      "struct": {
        "fullyQualifiedName": "Types.std_msgs_Header"
      }
    },
    {
      "struct": {
        "fullyQualifiedName": "Types.builtin_interfaces_Time"
      }
    },
    {
      "property": {
        "fullyQualifiedName": "Types.builtin_interfaces_Time.sec",
        "dataType": "INT32",
        "dataEncoding": "TYPED"
      }
    },
    {
      "property": {
        "fullyQualifiedName": "Types.builtin_interfaces_Time.nanosec",
        "dataType": "UINT32",
        "dataEncoding": "TYPED"
      }
    }
  ]
}
```

```
}
},
{
  "property": {
    "fullyQualifiedName": "Types.std_msgs_Header.stamp",
    "dataType": "STRUCT",
    "structFullyQualifiedName": "Types.builtin_interfaces_Time"
  }
},
{
  "property": {
    "fullyQualifiedName": "Types.std_msgs_Header.frame_id",
    "dataType": "STRING",
    "dataEncoding": "TYPED"
  }
},
{
  "property": {
    "fullyQualifiedName": "Types.sensor_msgs_msg_CompressedImage.header",
    "dataType": "STRUCT",
    "structFullyQualifiedName": "Types.std_msgs_Header"
  }
},
{
  "property": {
    "fullyQualifiedName": "Types.sensor_msgs_msg_CompressedImage.format",
    "dataType": "STRING",
    "dataEncoding": "TYPED"
  }
},
{
  "property": {
    "fullyQualifiedName": "Types.sensor_msgs_msg_CompressedImage.data",
    "dataType": "UINT8_ARRAY",
    "dataEncoding": "BINARY"
  }
},
{
  "branch": {
    "fullyQualifiedName": "Vehicle",
    "description": "Vehicle"
  }
},
{
```

```
"branch": {
  "fullyQualifiedName": "Vehicle.Cameras"
},
{
  "branch": {
    "fullyQualifiedName": "Vehicle.Cameras.Front"
  },
  {
    "sensor": {
      "fullyQualifiedName": "Vehicle.Cameras.Front.Image",
      "dataType": "STRUCT",
      "structFullyQualifiedName": "Types.sensor_msgs_msg_CompressedImage"
    },
    {
      "struct": {
        "fullyQualifiedName": "Types.std_msgs_msg_Float64"
      },
      {
        "property": {
          "fullyQualifiedName": "Types.std_msgs_msg_Float64.data",
          "dataType": "DOUBLE",
          "dataEncoding": "TYPED"
        },
        {
          "sensor": {
            "fullyQualifiedName": "Vehicle.Velocity",
            "dataType": "STRUCT",
            "structFullyQualifiedName": "Types.std_msgs_msg_Float64"
          },
          {
            "struct": {
              "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest"
            },
            {
              "property": {
                "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest.x_offset",
                "dataType": "UINT32",
```

```
    "dataEncoding": "TYPED"
  }
},
{
  "property": {
    "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest.y_offset",
    "dataType": "UINT32",
    "dataEncoding": "TYPED"
  }
},
{
  "property": {
    "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest.height",
    "dataType": "UINT32",
    "dataEncoding": "TYPED"
  }
},
{
  "property": {
    "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest.width",
    "dataType": "UINT32",
    "dataEncoding": "TYPED"
  }
},
{
  "property": {
    "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest.do_rectify",
    "dataType": "BOOLEAN",
    "dataEncoding": "TYPED"
  }
},
{
  "branch": {
    "fullyQualifiedName": "Vehicle.Perception"
  }
},
{
  "sensor": {
    "fullyQualifiedName": "Vehicle.Perception.Obstacle",
    "dataType": "STRUCT",
    "structFullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest"
  }
}
]
```

```
}
```

Note

Você pode baixar um [script de demonstração](#) para converter ROS 2 mensagens em VSS JSON arquivos compatíveis com o catálogo de sinais. Para ter mais informações, consulte o [Vision System Data Developer Guide](#).

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Importar um catálogo AWS de sinais de IoT FleetWise

Você pode usar o FleetWise console de AWS IoT ou API importar um catálogo de sinais.

Tópicos

- [Importar um catálogo de sinais \(console\)](#)
- [Importar um catálogo de sinais \(AWS CLI\)](#)

Importar um catálogo de sinais (console)

Você pode usar o FleetWise console de AWS IoT para importar um catálogo de sinais.

Important

Você pode ter no máximo um catálogo de sinais. Se você já tem um catálogo de sinais, não verá a opção de importar um catálogo de sinais no console.

Para importar um catálogo de sinais

1. Abra o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Catálogo de sinais.
3. Na página de resumo do catálogo de sinais, selecione Importar catálogo de sinais.
4. Importe o arquivo que contém os sinais.
 - Para fazer upload de um arquivo de um bucket S3:
 - a. Escolha Import from S3 (Importar do S3).

- b. Selecione Browse S3 (Navegar no S3).
- c. Em Buckets, insira o nome do bucket ou objeto, selecione-o na lista e escolha o arquivo na lista. Selecione o botão Escolher arquivo.

Ou, para o S3 URI, insira um Amazon Simple Storage ServiceURI. Para obter mais informações, consulte [Métodos de acesso a um bucket](#) no Guia do usuário do Amazon S3.

- Para fazer upload de um arquivo do seu computador:
 - a. Selecione Importar arquivo.
 - b. Faça upload de um arquivo.json em formato de [Especificação de Sinal de Veículo \(VSS\)](#).

5. Verifique o catálogo de sinais e selecione Importar arquivo.

Importar um catálogo de sinais (AWS CLI)

Você pode usar a [ImportSignalCatalog](#) API operação para carregar um JSON arquivo que ajuda a criar um catálogo de sinais. Você deve seguir a [Especificação do Sinal do Veículo \(VSS\)](#) para salvar os sinais no JSON arquivo. O exemplo a seguir usa AWS CLI.

Para importar um catálogo de sinais, execute o comando a seguir.

- Substituir *signal-catalog-name* com o nome do catálogo de sinais que você está criando.
- (Opcional) Substitua a descrição por uma *description* para ajudá-lo a identificar o catálogo de sinais.
- Substituir *signal-catalog-configuration-vss* com o nome do arquivo de JSON string que contém sinais definidos em VSS.

Para obter mais informações sobre como configurar ramificações, atributos, sensores e atuadores, consulte [Configurar AWS sinais de IoT FleetWise](#) .

```
aws iotfleetwise import-signal-catalog \  
    --name signal-catalog-name \  
    --description description \  
    --vss file://signal-catalog-configuration-vss.json
```

Eles JSON devem ser sequenciados e passados pelo campo. `vssJson` A seguir está um exemplo de sinais definidos emVSS.

```
{
  "Vehicle": {
    "type": "branch",
    "children": {
      "Chassis": {
        "type": "branch",
        "description": "All data concerning steering, suspension, wheels, and brakes.",
        "children": {
          "SteeringWheel": {
            "type": "branch",
            "description": "Steering wheel signals",
            "children": {
              "Diameter": {
                "type": "attribute",
                "description": "The diameter of the steering wheel",
                "datatype": "float",
                "unit": "cm",
                "min": 1,
                "max": 50
              },
              "HandsOff": {
                "type": "branch",
                "children": {
                  "HandsOffSteeringState": {
                    "type": "actuator",
                    "description": "HndsOffStrWhlDtSt. Hands Off Steering State",
                    "datatype": "boolean"
                  },
                  "HandsOffSteeringMode": {
                    "type": "actuator",
                    "description": "HndsOffStrWhlDtMd. Hands Off Steering Mode",
                    "datatype": "int8",
                    "min": 0,
                    "max": 2
                  }
                }
              }
            }
          },
          "Accelerator": {
```

```

    "type": "branch",
    "description": "",
    "children": {
      "AcceleratorPedalPosition": {
        "type": "sensor",
        "description": "Throttle__Position. Accelerator pedal position as percent. 0 =
Not depressed. 100 = Fully depressed.",
        "datatype": "uint8",
        "unit": "%",
        "min": 0,
        "max": 100.000035
      }
    }
  },
  "Powertrain": {
    "type": "branch",
    "description": "Powertrain data for battery management, etc.",
    "children": {
      "Transmission": {
        "type": "branch",
        "description": "Transmission-specific data, stopping at the drive shafts.",
        "children": {
          "VehicleOdometer": {
            "type": "sensor",
            "description": "Vehicle_Odometer",
            "datatype": "float",
            "unit": "km",
            "min": 0,
            "max": 67108863.984375
          }
        }
      }
    }
  },
  "CombustionEngine": {
    "type": "branch",
    "description": "Engine-specific data, stopping at the bell housing.",
    "children": {
      "Engine": {
        "type": "branch",
        "description": "Engine description",
        "children": {
          "timing": {
            "type": "branch",

```



```
    "description": "timing description",
    "children": {
      "run_time": {
        "type": "sensor",
        "description": "Engine run time",
        "datatype": "int16",
        "unit": "ms",
        "min": 0,
        "max": 10000
      },
      "idle_time": {
        "type": "sensor",
        "description": "Engine idle time",
        "datatype": "int16",
        "min": 0,
        "unit": "ms",
        "max": 10000
      }
    }
  }
}
},
"Axle": {
  "type": "branch",
  "description": "Axle signals",
  "children": {
    "TireRRPrs": {
      "type": "sensor",
      "description": "TireRRPrs. Right rear Tire pressure in kilo-Pascal",
      "datatype": "float",
      "unit": "kPaG",
      "min": 0,
      "max": 1020
    }
  }
}
},
"Cameras": {
  "type": "branch",
```

```
"description": "Branch to aggregate all cameras in the vehicle",
"children": {
  "FrontViewCamera": {
    "type": "sensor",
    "datatype": "VehicleDataTypes.SVMCamera",
    "description": "Front view camera"
  },
  "RearViewCamera": {
    "type": "sensor",
    "datatype": "VehicleDataTypes.SVMCamera",
    "description": "Rear view camera"
  },
  "LeftSideViewCamera": {
    "type": "sensor",
    "datatype": "VehicleDataTypes.SVMCamera",
    "description": "Left side view camera"
  },
  "RightSideViewCamera": {
    "type": "sensor",
    "datatype": "VehicleDataTypes.SVMCamera",
    "description": "Right side view camera"
  }
}
},
"ComplexDataTypes": {
  "VehicleDataTypes": {
    "type": "branch",
    "description": "Branch to aggregate all camera related higher order data types",
    "children": {
      "SVMCamera": {
        "type": "struct",
        "description": "This data type represents Surround View Monitor (SVM) camera system in a vehicle",
        "comment": "Test comment",
        "deprecation": "Test deprecation message",
        "children": {
          "Make": {
            "type": "property",
            "description": "Make of the SVM camera",
            "datatype": "string",
            "comment": "Test comment",
            "deprecation": "Test deprecation message"
          },
          "Description": {
```

```
    "type": "property",
    "description": "Description of the SVM camera",
    "datatype": "string",
    "comment": "Test comment",
    "deprecation": "Test deprecation message"
  },
  "FPS": {
    "type": "property",
    "description": "FPS of the SVM camera",
    "datatype": "double",
    "comment": "Test comment",
    "deprecation": "Test deprecation message"
  },
  "Orientation": {
    "type": "property",
    "description": "Orientation of the SVM camera",
    "datatype": "VehicleDataTypes.Orientation",
    "comment": "Test comment",
    "deprecation": "Test deprecation message"
  },
  "Range": {
    "type": "property",
    "description": "Range of the SVM camera",
    "datatype": "VehicleDataTypes.Range",
    "comment": "Test comment",
    "deprecation": "Test deprecation message"
  },
  "RawData": {
    "type": "property",
    "description": "Represents binary data of the SVM camera",
    "datatype": "uint8[]",
    "dataencoding": "binary",
    "comment": "Test comment",
    "deprecation": "Test deprecation message"
  },
  "CapturedFrames": {
    "type": "property",
    "description": "Represents selected frames captured by the SVM camera",
    "datatype": "VehicleDataTypes.Frame[]",
    "dataencoding": "typed",
    "comment": "Test comment",
    "deprecation": "Test deprecation message"
  }
}
```

```
  },
  "Range": {
    "type": "struct",
    "description": "Range of a camera in centimeters",
    "comment": "Test comment",
    "deprecation": "Test deprecation message",
    "children": {
      "Min": {
        "type": "property",
        "description": "Minimum range of a camera in centimeters",
        "datatype": "uint32",
        "comment": "Test comment",
        "deprecation": "Test deprecation message"
      },
      "Max": {
        "type": "property",
        "description": "Maximum range of a camera in centimeters",
        "datatype": "uint32",
        "comment": "Test comment",
        "deprecation": "Test deprecation message"
      }
    }
  },
  "Orientation": {
    "type": "struct",
    "description": "Orientation of a camera",
    "comment": "Test comment",
    "deprecation": "Test deprecation message",
    "children": {
      "Front": {
        "type": "property",
        "description": "Indicates whether the camera is oriented to the front of the
vehicle",
        "datatype": "boolean",
        "comment": "Test comment",
        "deprecation": "Test deprecation message"
      },
      "Rear": {
        "type": "property",
        "description": "Indicates whether the camera is oriented to the rear of the
vehicle",
        "datatype": "boolean",
        "comment": "Test comment",
        "deprecation": "Test deprecation message"
      }
    }
  }
}
```



```
{
  "type": "actuator",
  "description": "HndsOffStrWhlDtMd. Hands Off Steering Mode",
  "datatype": "int8",
  "min": 0,
  "max": 2
},
{
  "type": "branch",
  "description": "",
  "children": {
    "AcceleratorPedalPosition": {
      "type": "sensor",
      "description": "Throttle__Position. Accelerator pedal position as percent. 0 = Not depressed. 100 = Fully depressed.",
      "datatype": "uint8",
      "unit": "%",
      "min": 0,
      "max": 100.000035
    }
  }
},
{
  "type": "branch",
  "description": "Powertrain data for battery management, etc.",
  "children": {
    "Transmission": {
      "type": "branch",
      "description": "Transmission-specific data, stopping at the drive shafts.",
      "children": {
        "VehicleOdometer": {
          "type": "sensor",
          "description": "Vehicle_Odometer",
          "datatype": "float",
          "unit": "km",
          "min": 0,
          "max": 67108863.984375
        }
      }
    },
    "CombustionEngine": {
      "type": "branch",
      "description": "Engine-specific data, stopping at the bell housing.",
      "children": {
        "Engine": {
          "type": "branch",
          "description": "Engine description",
          "children": {
            "timing": {
              "type": "branch",
              "description": "timing description",
              "children": {
                "run_time": {
                  "type": "sensor",
                  "description": "Engine run time",
                  "datatype": "int16",
                  "unit": "ms",
                  "min": 0,
                  "max": 10000
                },
                "idle_time": {
                  "type": "sensor",
                  "description": "Engine idle time",
                  "datatype": "int16",
                  "min": 0,
                  "unit": "ms",
                  "max": 10000
                }
              }
            }
          }
        }
      }
    }
  }
},
{
  "type": "branch",
  "description": "Axle signals",
  "children": {
    "TireRRPrs": {
      "type": "sensor",
      "description": "TireRRPrs. Right rear Tire pressure in kilo-Pascal",
      "datatype": "float",
      "unit": "kPaG",
      "min": 0,
      "max": 1020
    }
  }
}
}
```

Note

Você pode baixar um [script de demonstração](#) para converter ROS 2 mensagens em VSS JSON arquivos compatíveis com o catálogo de sinais. Para ter mais informações, consulte o [Vision System Data Developer Guide](#).

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Atualizar um catálogo AWS de sinais de IoT FleetWise

Você pode usar a [UpdateSignalCatalog](#) API operação para atualizar um catálogo de sinais existente. O exemplo a seguir usa AWS CLI.

Para atualizar um catálogo de sinais existente, execute o comando a seguir.

Substituir *signal-catalog-configuration* com o nome do JSON arquivo que contém a configuração.

```
aws iotfleetwise update-signal-catalog --cli-input-json file://signal-catalog-configuration.json
```

Substituir *signal-catalog-name* com o nome do catálogo de sinais que você está atualizando.

Para obter mais informações sobre como configurar ramificações, atributos, sensores e atuadores, consulte [Configurar AWS sinais de IoT FleetWise](#).

Important

Estruturas personalizadas são imutáveis. Se você precisar reordenar ou inserir propriedades em uma estrutura personalizada (struct) existente, exclua a estrutura e crie uma totalmente nova com a ordem de propriedades desejada.

Para excluir uma estrutura personalizada, adicione o nome totalmente qualificado da estrutura em `nodesToRemove`. Uma estrutura não poderá ser excluída se for referida por algum sinal. Todos os sinais que se referem à estrutura (seu tipo de dados é definido como a estrutura de destino) devem ser atualizados ou excluídos antes da solicitação de atualização do catálogo de sinais.

```
{
  "name": "signal-catalog-name",
  "nodesToAdd": [{
    "branch": {
      "description": "Front left of vehicle specific data.",
      "fullyQualifiedName": "Vehicle.Front.Left"
    }
  },
  {
    "branch": {
      "description": "Door-specific data for the front left of vehicle.",
      "fullyQualifiedName": "Vehicle.Front.Left.Door"
    }
  },
  {
    "actuator": {
      "fullyQualifiedName": "Vehicle.Front.Left.Door.Lock",
      "description": "Whether the front left door is locked.",
      "dataType": "BOOLEAN"
    }
  }
}
```

```
    },
    {
      "branch": {
        "fullyQualified_name": "Vehicle.Camera"
      }
    },
    {
      "struct": {
        "fullyQualified_name": "Vehicle.Camera.SVMCamera"
      }
    },
    {
      "property": {
        "fullyQualified_name": "Vehicle.Camera.SVMCamera.ISO",
        "dataType": "STRING"
      }
    }
  ],
  "nodesToRemove": ["Vehicle.Chassis.SteeringWheel.HandsOffSteeringState"],
  "nodesToUpdate": [{
    "attribute": {
      "dataType": "FLOAT",
      "fullyQualified_name": "Vehicle.Chassis.SteeringWheel.Diameter",
      "max": 55
    }
  }]
}
```

Excluir um catálogo AWS de sinais de IoT FleetWise

Você pode usar a [DeleteSignalCatalog](#) API operação para excluir um catálogo de sinais. O exemplo a seguir usa AWS CLI.

Important

Antes de excluir um catálogo de sinais, verifique se ele não tem modelos de veículos, manifestos de decodificadores, veículos, frotas ou campanhas associados. Para obter instruções, consulte:

- [Excluir um modelo AWS de veículo de IoT FleetWise](#)
- [Excluir um manifesto AWS do decodificador de IoT FleetWise](#)
- [Excluir um veículo AWS de IoT FleetWise](#)

- [Excluir uma frota AWS de IoT FleetWise](#)
- [Excluir uma campanha AWS de IoT FleetWise](#)

Para excluir um catálogo de sinais existente, execute o comando a seguir. Substituir *signal-catalog-name* com o nome do catálogo de sinais que você está excluindo.

```
aws iotfleetwise delete-signal-catalog --name signal-catalog-name
```

Note

Esse comando não gera resultados.

Obtenha informações AWS do catálogo de FleetWise sinais de IoT

Você pode usar a [ListSignalCatalogs](#) API operação para verificar se um catálogo de sinais foi excluído. O exemplo a seguir usa AWS CLI.

Para recuperar uma lista paginada de resumos de todos os catálogos de sinais, execute o seguinte comando.

```
aws iotfleetwise list-signal-catalogs
```

Você pode usar a [ListSignalCatalogNodes](#) API operação para verificar se um catálogo de sinais foi atualizado. O exemplo a seguir usa AWS CLI.

Para recuperar uma lista paginada de resumos de todos os sinais (nós) em um determinado catálogo de sinais, execute o seguinte comando.

Substituir *signal-catalog-name* com o nome do catálogo de sinais que você está verificando.

```
aws iotfleetwise list-signal-catalog-nodes --name signal-catalog-name
```

Você pode usar a [GetSignalCatalog](#) API operação para recuperar informações do catálogo de sinais. O exemplo a seguir usa AWS CLI.

Para recuperar informações sobre um catálogo de sinais, execute o comando a seguir.

Substituir *signal-catalog-name* com o nome do catálogo de sinais que você deseja recuperar.

```
aws iotfleetwise get-signal-catalog --name signal-catalog-name
```

Note

Essa operação é eventualmente consistente. Em outras palavras, as alterações no catálogo de sinais podem não ser refletidas imediatamente.

Gerencie modelos AWS de veículos de IoT FleetWise

Você usa sinais para criar modelos de veículos que ajudam a padronizar o formato de seus veículos. Os modelos de veículos impõem informações consistentes em vários veículos do mesmo tipo, para que você possa processar dados de frotas de veículos. Veículos criados com base no mesmo modelo de veículo herdam o mesmo grupo de sinais. Para obter mais informações, consulte [Gerencie AWS veículos de IoT FleetWise](#).

Cada modelo de veículo tem um campo de status que contém o estado do modelo do veículo. O estado pode ser um dos seguintes valores:

- ACTIVE — O modelo do veículo está ativo.
- DRAFT — A configuração do modelo do veículo será salva.

⚠ Important

- Se você quiser usar a `CreateModelManifest` API operação para criar o primeiro modelo de veículo, primeiro deverá criar um catálogo de sinais. Para obter mais informações, consulte [Crie um catálogo de AWS sinais de IoT FleetWise](#).
- Se você usa o FleetWise console de AWS IoT para criar um modelo de veículo, a AWS IoT ativa FleetWise automaticamente o modelo de veículo para você.
- Se você usar a `CreateModelManifest` API operação para criar um modelo de veículo, o modelo do veículo permanecerá no DRAFT estado.
- Você não pode criar veículos usando modelos que estejam no estado DRAFT. Use a `UpdateModelManifest` API operação para alterar os modelos de veículos para o ACTIVE estado.

- Não é possível editar modelos de veículos que estejam no estado ACTIVE.

Tópicos

- [Crie um modelo de AWS veículo de IoT FleetWise](#)
- [Atualizar um modelo AWS de veículo de IoT FleetWise](#)
- [Excluir um modelo AWS de veículo de IoT FleetWise](#)
- [Obtenha informações sobre modelos de FleetWise veículos AWS de IoT](#)

Crie um modelo de AWS veículo de IoT FleetWise

Você pode usar o FleetWise console de AWS IoT ou API criar modelos de veículos.

Important

Você deve ter um catálogo de sinais antes de criar um modelo de veículo usando a `CreateModelManifest` API operação.

Tópicos

- [Criar um modelo de veículo \(console\)](#)
- [Criar um modelo de veículo \(AWS CLI\)](#)

Criar um modelo de veículo (console)

No FleetWise console de AWS IoT, você pode criar um modelo de veículo das seguintes formas:

- [Use um modelo fornecido por AWS](#)
- [Criar manualmente um modelo de veículo](#)
- [Duplicar um modelo de veículo](#)

Use um modelo fornecido por AWS

AWS FleetWise A IoT fornece um modelo de diagnóstico integrado (OBD) II, J1979 que cria automaticamente um catálogo de sinais, um modelo de veículo e um manifesto de decodificador para

você. O modelo também adiciona interfaces OBD de rede ao manifesto do decodificador. Para obter mais informações, consulte [Gerencie manifestos AWS de decodificadores de IoT FleetWise](#).

Para criar um modelo de veículo usando um modelo

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Modelos de veículos.
3. Na página Modelos de veículos, selecione Adicionar modelo fornecido.
4. Escolha Diagnóstico integrado (OBD) II.
5. Insira um nome para a interface OBD de rede que a AWS IoT FleetWise está criando.
6. Escolha Adicionar.

Criar manualmente um modelo de veículo

Você pode adicionar sinais do catálogo de sinais ou importar sinais fazendo upload de um ou mais arquivos .dbc. Um arquivo.dbc é um formato de arquivo suportado pelos bancos de dados do Controller Area Network (CANbus).

Important

Você não pode criar um modelo de veículo com sinais de dados do sistema de visão usando o console de AWS IoT FleetWise. Em vez disso, use o AWS CLI para criar um modelo de veículo.

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Para criar manualmente um modelo de veículo

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Modelos de veículos.
3. Na página Modelos de veículos, selecione Criar modelo de veículo e faça o seguinte.

Tópicos

- [Etapa 1: configurar o modelo do veículo](#)
- [Etapa 2: adicionar sinais](#)

- [Etapa 3: importar sinais](#)
- [\(Opcional\) Etapa 4: adicionar atributos](#)
- [Etapa 5: analisar e criar](#)

Etapa 1: configurar o modelo do veículo

Em Informações gerais, faça o seguinte:

1. Insira um nome para o modelo do veículo.
2. (Opcional) Insira uma descrição.
3. Escolha Próximo.

Etapa 2: adicionar sinais

Note

- Se for a primeira vez que você usa a AWS IoT FleetWise, essa etapa não estará disponível até que você tenha um catálogo de sinais. Quando o primeiro modelo de veículo é criado, a AWS IoT cria FleetWise automaticamente um catálogo de sinais com sinais adicionados ao primeiro modelo de veículo.
- Se você tem experiência com AWS IoT FleetWise, pode adicionar sinais ao modelo do seu veículo selecionando sinais do catálogo de sinais ou carregando arquivos.dbc para importar sinais.
- Você deve ter pelo menos um sinal para criar um modelo de veículo.

Para adicionar sinais

1. Escolha um ou mais sinais do catálogo de sinais que você está adicionando ao modelo do veículo. Você pode revisar os sinais selecionados no painel direito.

Note

Apenas os sinais selecionados serão adicionados ao modelo do veículo.

2. Escolha Próximo.

Etapa 3: importar sinais

Note

- Se for a primeira vez que você usa a AWS IoT FleetWise, você deve carregar pelo menos um arquivo.dbc para importar sinais.
- Se você tem experiência com AWS IoT FleetWise, pode adicionar sinais ao modelo do seu veículo selecionando sinais do catálogo de sinais ou carregando arquivos.dbc para importar sinais.
- Você deve ter pelo menos um sinal para criar um modelo de veículo.

Para importar sinais

1. Selecione Selecionar arquivos.
2. Na caixa de diálogo, escolha o arquivo .dbc que contém sinais. Você pode fazer upload de vários arquivos .dbc.
3. AWS A IoT FleetWise analisa seus arquivos.dbc para recuperar sinais.

Na seção Sinais, especifique os seguintes metadados para cada sinal.

- Name — Nome do sinal.

O nome do sinal deve ser exclusivo. O nome do sinal mais o caminho podem ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e _ (sublinhado).

- Tipo de dados — O tipo de dados do sinal deve ser um dos seguintes: INT8, UIN8, INT16, UIN16,, INT32, UIN32, INT64, UIN64, BOOLEAN, FLOAT, UNIX _DOUBLESTRING, _TIMESTAMP, INT8 _ARRAY, UIN8 _ARRAY, INT16 _ARRAY, UIN16 _ARRAY, INT32 _ARRAY, _, UIN32 _ARRAY, INT64 _ARRAY, UIN64 _ARRAY, BOOLEAN _ARRAY, FLOAT _ARRAY, DOUBLE _ARRAY, STRING_ ARRAYARRAY, UNIX TIMESTAMP __ ou UNKNOWN.
- Signal type — O tipo de sinal, que pode ser Sensor ou Atuador.
- (Opcional) Unit — A unidade científica do sinal, como km ou Celsius.
- (Opcional) Path — O caminho para o sinal. Da mesma forma JSONPath, use um ponto (.) para se referir ao sinal de uma criança. Por exemplo, **Vehicle.Engine.Light**.

O nome do sinal mais o caminho podem ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e _ (sublinhado).

- (Opcional) Min — O valor mínimo do sinal.
- (Opcional) Max — O valor máximo do sinal.
- (Opcional) Description — A descrição do sinal.

A descrição pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), _ (sublinhado) e - (hífen).

4. Escolha Próximo.

(Opcional) Etapa 4: adicionar atributos

É possível adicionar até 100 atributos, incluindo os atributos existentes no catálogo de sinais.

Para adicionar atributos

1. Em Adicionar atributos, especifique os seguintes metadados para cada atributo.

- Name — Nome do atributo.

O nome do sinal deve ser exclusivo. O nome e o caminho do sinal podem ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e _ (sublinhado)

- Tipo de dados — O tipo de dados do atributo deve ser um dos seguintes:
INT8,UINT8,INT16,UINT16,,INT32,UINT32,INT64,UINT64,BOOLEAN,FLOAT, UNIX _
DOUBLESTRING, _TIMESTAMP, INT8 _ARRAY, UINT8 _ARRAY, INT16 _ARRAY, UINT16
_ARRAY, INT32 _ARRAY, _, UINT32 _ARRAY, INT64 _ARRAY, UINT64 _ARRAY, BOOLEAN
_ARRAY, FLOAT _ARRAY, DOUBLE _ARRAY, STRING_ARRAY, UNIX TIMESTAMP _
ARRAY ou UNKNOWN
- (Opcional) Unit — A unidade científica do atributo, como km ou Celsius.
- (Opcional) Path — O caminho para o sinal. Da mesma formaJSONPath, use um ponto (.) para se referir ao sinal de uma criança. Por exemplo, **Vehicle.Engine.Light**.

O nome do sinal mais o caminho podem ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e _ (sublinhado)

- (Opcional) Min — O valor mínimo do atributo.
- (Opcional) Max — O valor máximo do atributo.

- (Opcional) Description — A descrição do atributo.

A descrição pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), _ (sublinhado) e - (hífen).

2. Escolha Próximo.

Etapa 5: analisar e criar

Verifique as configurações do modelo do veículo e selecione Criar.

Duplicar um modelo de veículo

AWS A IoT FleetWise pode copiar as configurações de um modelo de veículo existente para criar um novo modelo. Os sinais especificados no modelo de veículo selecionado são copiados para o novo modelo de veículo.

Para duplicar um modelo de veículo

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Modelos de veículos.
3. Selecione um modelo na lista de modelos de veículos e depois Duplicar modelo.

Para configurar o modelo do veículo, siga o tutorial [Criar manualmente um modelo de veículo](#).

Pode levar alguns minutos para que a AWS IoT processe sua solicitação FleetWise para criar o modelo do veículo. Depois que o modelo do veículo for criado com sucesso, na página Modelos do veículo, a coluna Status é exibida ACTIVE. Quando o modelo do veículo se torna ativo, não é possível editá-lo.

Criar um modelo de veículo (AWS CLI)

Você pode usar a [CreateModelManifest](#) API operação para criar modelos de veículos (manifestos de modelos). O exemplo a seguir usa a AWS CLI.

⚠ Important

Se você quiser usar a AWS IoT FleetWise API para criar o primeiro modelo de veículo, primeiro deverá criar um catálogo de sinais. Para obter mais informações sobre como criar um catálogo de sinais, consulte [Crie um catálogo de AWS sinais de IoT FleetWise](#).

Para criar um modelo de veículo, execute o seguinte comando.

Substituir *vehicle-model-configuration* com o nome do JSON arquivo que contém a configuração.

```
aws iotfleetwise create-model-manifest --cli-input-json file://vehicle-model-configuration.json
```

- Substituir *vehicle-model-name* com o nome do modelo do veículo que você está criando.
- Substituir *signal-catalog-ARN* com o Amazon Resource Name (ARN) do catálogo de sinais.
- (Opcional) Substituir *description* com uma descrição para ajudá-lo a identificar o modelo do veículo.

Para obter mais informações sobre como configurar ramificações, atributos, sensores e atuadores, consulte [Configurar AWS sinais de IoT FleetWise](#).

```
{  
  "name": "vehicle-model-name",  
  "signalCatalogArn": "signal-catalog-ARN",  
  "description": "description",  
  "nodes": ["Vehicle.Chassis"]  
}
```

Atualizar um modelo AWS de veículo de IoT FleetWise

Você pode usar a [UpdateModelManifest](#) API operação para atualizar um modelo de veículo existente (manifestos de modelo). O exemplo a seguir usa a AWS CLI.

Para atualizar um modelo de veículo existente, execute o seguinte comando.

Substituir *update-vehicle-model-configuration* com o nome do JSON arquivo que contém a configuração.

```
aws iotfleetwise update-model-manifest --cli-input-json file://update-vehicle-model-configuration.json
```

- Substituir *vehicle-model-name* com o nome do modelo do veículo que você está atualizando.
- (Opcional) Para ativar o modelo do veículo, substitua *vehicle-model-status* com ACTIVE.

Important

Após a ativação do modelo do veículo, você não poderá alterá-lo.

- (Opcional) Substituir *description* com uma descrição atualizada para ajudá-lo a identificar o modelo do veículo.

```
{
  "name": "vehicle-model-name",
  "status": "vehicle-model-status",
  "description": "description",
  "nodesToAdd": ["Vehicle.Front.Left"],
  "nodesToRemove": ["Vehicle.Chassis.SteeringWheel"],
}
```

Excluir um modelo AWS de veículo de IoT FleetWise

Você pode usar o FleetWise console de AWS IoT ou excluir modelos API de veículos.

Important

Os veículos e os manifestos do decodificador associados ao modelo do veículo devem ser excluídos primeiro. Para ter mais informações, consulte [Excluir um veículo AWS de IoT FleetWise](#) e [Excluir um manifesto AWS do decodificador de IoT FleetWise](#).

Excluir um modelo de veículo (console)

Para excluir um modelo de veículo, use o console de AWS IoT. FleetWise

Para excluir um modelo de veículo

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Modelos de veículos.
3. Na página Modelos de veículos, escolha o modelo do veículo de destino.
4. Escolha Excluir.
5. Em Excluir **vehicle-model-name?**, insira o nome do modelo do veículo a ser excluído e escolha Confirmar.

Excluir um modelo de veículo (AWS CLI)

Você pode usar a [DeleteModelManifest](#) API operação para excluir um modelo de veículo existente (manifestos de modelo). O exemplo a seguir usa a AWS CLI.

Para excluir um modelo de veículo, execute o seguinte comando.

Substituir *model-manifest-name* com o nome do modelo do veículo que você está excluindo.

```
aws iotfleetwise delete-model-manifest --name model-manifest-name
```

Note

Esse comando não gera resultados.

Obtenha informações sobre modelos de FleetWise veículos AWS de IoT

Você pode usar a [ListModelManifests](#) API operação para verificar se um modelo de veículo foi excluído. O exemplo a seguir usa AWS CLI.

Para recuperar uma lista paginada de resumos de todos os modelos de veículos, execute o seguinte comando.

```
aws iotfleetwise list-model-manifests
```

Você pode usar a [ListModelManifestNodes](#) API operação para verificar se um modelo de veículo foi atualizado. O exemplo a seguir usa AWS CLI.

Para recuperar uma lista paginada de resumos de todos os sinais (nós) em um determinado modelo de veículo, execute o seguinte comando.

Substituir *vehicle-model-name* com o nome do modelo do veículo que você está verificando.

```
aws iotfleetwise list-model-manifest-nodes /  
    --name vehicle-model-name
```

Para recuperar informações sobre um modelo de veículo, execute o comando a seguir.

Substituir *vehicle-model* com o nome do modelo do veículo que você deseja recuperar.

```
aws iotfleetwise get-model-manifest --name vehicle-model
```

Note

Essa operação é [eventualmente consistente](#). Em outras palavras, as alterações ao modelo do veículo podem não ser refletidas imediatamente.

Gerencie manifestos AWS de decodificadores de IoT FleetWise

Os manifestos do decodificador contêm informações de decodificação que a AWS IoT FleetWise usa para transformar dados do veículo (dados binários) em valores legíveis por humanos e preparar seus dados para análises de dados. A interface de rede e os sinais do decodificador são os principais componentes com os quais você trabalha para configurar os manifestos do decodificador.

Interface de rede

Contém informações sobre o protocolo que a rede do veículo usa. AWS A IoT é FleetWise compatível com os seguintes protocolos.

Rede de área do controlador (CANbus)

Um protocolo que define como os dados são comunicados entre unidades de controle eletrônico (ECUs). ECUs pode ser a unidade de controle do motor, os airbags ou o sistema de áudio.

Diagnóstico a bordo (IIOBD)

Um protocolo mais desenvolvido que define como os dados de autodiagnóstico são comunicados entre eles. ECUs Ele fornece vários códigos de problemas de diagnóstico padrão (DTCs) que ajudam a identificar o que há de errado com seu veículo.

Middleware de veículo

O middleware de veículo definido como um tipo de interface de rede. Exemplos de middleware de veículos incluem Robot Operating System (ROS2) e Scalable Service-Oriented Middleware over IP (/IP). SOME

Note

AWS A IoT FleetWise oferece suporte a ROS 2 middlewares para dados do sistema de visão.

Sinal do decodificador

Fornece informações detalhadas de decodificação para um sinal específico. Cada sinal especificado no modelo do veículo deve ser emparelhado com um sinal de decodificador. Se o manifesto do decodificador contiver interfaces CAN de rede, ele deverá conter sinais do CAN decodificador. Se o manifesto do decodificador contiver interfaces OBD de rede, ele deverá conter sinais do OBD decodificador.

O manifesto do decodificador deverá conter sinais do decodificador de mensagens se também contiver interfaces de middleware do veículo.

Cada manifesto do decodificador deve estar associado a um modelo de veículo. AWS A IoT FleetWise usa o manifesto do decodificador associado para decodificar dados de veículos criados com base no modelo do veículo.

Cada manifesto do decodificador tem um campo de status que contém o estado do manifesto do decodificador. O estado pode ser um dos seguintes valores:

- ACTIVE — O manifesto do decodificador está ativo.
- DRAFT — A configuração do manifesto do decodificador não é salva.

- **VALIDATING:** o manifesto do decodificador está em validação quanto a sua elegibilidade. Isso se aplica somente aos manifestos do decodificador que contêm pelo menos um sinal de dados de sistemas de visão.
- **INVALID:** o manifesto do decodificador falhou na validação e ainda não pode ser ativado. Isso se aplica somente aos manifestos do decodificador que contêm pelo menos um sinal de dados de sistemas de visão. Você pode usar o `ListDecoderManifests` e `GetDecoderManifest` APIs para verificar o motivo da falha na validação.

Important

- Se você usa o FleetWise console de AWS IoT para criar um manifesto do decodificador, a AWS IoT ativa FleetWise automaticamente o manifesto do decodificador para você.
- Se você usar a `CreateDecoderManifest` API operação para criar um manifesto do decodificador, o manifesto do decodificador permanecerá no DRAFT estado.
- Você não pode criar veículos usando modelos de veículos associados a um manifesto do decodificador da DRAFT. Use a `UpdateDecoderManifest` API operação para alterar o manifesto do decodificador para o ACTIVE estado.
- Você não pode editar manifestos do decodificador que estejam no estado ACTIVE.

Tópicos

- [Configurar interfaces de FleetWise rede AWS de IoT e sinais de decodificador](#)
- [Crie um manifesto AWS do decodificador de IoT FleetWise](#)
- [Atualizar um manifesto AWS do decodificador de IoT FleetWise](#)
- [Excluir um manifesto AWS do decodificador de IoT FleetWise](#)
- [Obtenha informações sobre o manifesto do FleetWise decodificador AWS de IoT](#)

Configurar interfaces de FleetWise rede AWS de IoT e sinais de decodificador

Cada manifesto de decodificador tem pelo menos uma interface de rede e sinais de decodificador emparelhados com sinais especificados no modelo de veículo associado.

Se o manifesto do decodificador contiver interfaces CAN de rede, ele deverá conter sinais do CAN decodificador. Se o manifesto do decodificador contiver interfaces OBD de rede, ele deverá conter sinais do OBD decodificador.

Tópicos

- [Configurar interfaces de rede](#)
- [Configurar sinais do decodificador](#)

Configurar interfaces de rede

Para configurar uma interface CAN de rede, especifique as informações a seguir.

- `name`— O nome da CAN interface.

O nome da interface deve ser exclusivo e pode ter de 1 a 100 caracteres.

- (Opcional) `protocolName` — O nome do protocolo.

Valores válidos: CAN-FD e CAN

- (Opcional) `protocolVersion` — FleetWise Atualmente, a AWS IoT suporta CAN -FD e 2.0b. CAN

Valores válidos: 1.0 e 2.0b

Para configurar uma interface OBD de rede, especifique as informações a seguir.

- `name`— O nome da OBD interface.

O nome da interface deve ser exclusivo e pode ter de 1 a 100 caracteres.

- `requestMessageId` — O ID da mensagem que está solicitando dados.
- (Opcional) `dtcRequestIntervalSeconds` — Com que frequência solicitar códigos de problemas de diagnóstico (DTCs) do veículo em segundos. Por exemplo, se o valor especificado for 120, o software Edge Agent coleta os arquivos armazenados DTCs uma vez a cada 2 minutos.
- (Opcional) `hasTransmissionEcu` — Se o veículo tem um módulo de controle de transmissão (TCM).

Valores válidos: `true` e `false`

- (Opcional) `obdStandard` — O OBD padrão compatível com a AWS IoT. FleetWise AWS FleetWise Atualmente, a IoT suporta o padrão World Wide Harmonization On-Board Diagnostics (WWH-) -4. OBD ISO15765
- (Opcional) `pidRequestIntervalSeconds` — Com que frequência solicitar OBD II PIDs do veículo. Por exemplo, se o valor especificado for 120, o software Edge Agent coletará OBD II PIDs uma vez a cada 2 minutos.
- (Opcional) `useExtendedIds` — Se deve usar estendido IDs na mensagem.

Valores válidos: `true` e `false`

Para configurar uma interface de rede de middleware, especifique as informações a seguir.

- `name`: o nome da interface de middleware do veículo.

O nome da interface deve ser exclusivo e pode ter de 1 a 100 caracteres.

- `protocolName`: o nome do protocolo.

Valores válidos: `ROS_2`

Configurar sinais do decodificador

Para configurar um sinal de CAN decodificador, especifique as informações a seguir.

- `factor` — O multiplicador usado para decodificar a mensagem.
- `isBigEndian` — Caso a ordem de bytes de uma mensagem seja big endian. Se for big endian, o valor mais significativo na sequência será armazenado primeiro, no endereço de armazenamento mais baixo.
- `isSigned` — Caso a mensagem esteja assinada. Se estiver assinada, a mensagem poderá representar números positivos e negativos.
- `length` — O comprimento da mensagem em bytes.
- `messageId` — O ID da mensagem.
- `offset` — O deslocamento usado para calcular o valor do sinal. Combinado com o fator, o cálculo é `value = raw_value * factor + offset`.
- `startBit` — Indica a localização do primeiro bit da mensagem.
- (Opcional) `name` — Nome do sinal.

Para configurar um sinal de OBD decodificador, especifique as informações a seguir.

- `byteLength` — O comprimento da mensagem em bytes.
- `offset` — O deslocamento usado para calcular o valor do sinal. Combinado com a escala, o cálculo é `value = raw_value * scaling + offset`.
- `pid` — O código de diagnóstico utilizado para solicitar uma mensagem de um veículo para este sinal.
- `pidResponseLength` — O tamanho da mensagem solicitada.
- `scaling` — O multiplicador utilizado para decodificar a mensagem.
- `serviceMode` — O modo de operação (serviço de diagnóstico) em uma mensagem.
- `startByte` — Indica o início da mensagem.
- (Opcional) `bitMaskLength` — O número de bits mascarados em uma mensagem.
- (Opcional) `bitRightShift` — O número de posições deslocadas para a direita.

Para configurar um sinal de decodificador de mensagens, especifique as informações a seguir.

- `topicName`: o nome do tópico para o sinal da mensagem. Corresponde aos tópicos em ROS 2. Para obter mais informações sobre o objeto de mensagem estruturada, consulte [StructuredMessage](#).
- `structuredMessage`: a mensagem estruturada para o sinal da mensagem. Ele pode ser definido com a `primitiveMessageDefinition`, `structuredMessageList` Definição ou `structuredMessageDefinition` recursivamente.

Crie um manifesto AWS do decodificador de IoT FleetWise

Você pode usar o FleetWise console de AWS IoT ou criar um API manifesto de decodificador para o modelo do seu veículo.

Important

Você deve ter um modelo de veículo antes de poder criar um manifesto do decodificador. Cada manifesto do decodificador deve estar associado a um modelo de veículo. Para obter mais informações, consulte [Gerencie modelos AWS de veículos de IoT FleetWise](#).

Tópicos

- [Criar um manifesto do decodificador \(console\)](#)
- [Criar um manifesto do decodificador \(AWS CLI\)](#)

Criar um manifesto do decodificador (console)

Você pode usar o FleetWise console de AWS IoT para criar um manifesto do decodificador associado ao modelo do seu veículo.

Important

Você não pode configurar sinais de dados do sistema de visão em manifestos de decodificadores usando o console de IoT AWS . FleetWise Use a AWS CLI em vez disso. O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Para criar um manifesto do decodificador

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Modelos de veículos.
3. Selecione o modelo de veículo de destino.
4. Na página de resumo do modelo do veículo, selecione Criar manifesto do decodificador e faça o seguinte.

Tópicos

- [Etapa 1: configurar o manifesto do decodificador](#)
- [Etapa 2: adicionar interfaces de rede](#)
- [Etapa 3: revisar e criar](#)

Etapa 1: configurar o manifesto do decodificador

Em Informações gerais, faça o seguinte:

1. Insira um nome exclusivo para o manifesto do decodificador.
2. (Opcional) Insira uma descrição.

3. Escolha Próximo.

Etapa 2: adicionar interfaces de rede

Cada manifesto do decodificador deve ter pelo menos uma interface de rede. Você pode adicionar várias interfaces de rede a um manifesto do decodificador.

Adicionar uma interface de rede

- Em Interface de rede, faça o seguinte.
 - a. Para o tipo de interface de rede, escolha CAN_ INTERFACE ou OBD_ INTERFACE.
 - b. Insira um nome exclusivo para sua interface de rede.
 - c. Insira um ID de interface de rede exclusivo. Você pode usar o ID gerado pela AWS IoT FleetWise.
 - d. Selecione um ou mais sinais especificados no modelo do seu veículo para emparelhar com sinais do decodificador.
 - e. Para fornecer informações de decodificação, faça upload de um arquivo .dbc. AWS A IoT FleetWise analisa o arquivo.dbc para recuperar sinais do decodificador.
 - f. Na seção Sinais emparelhados, verifique se cada sinal está emparelhado com um sinal de decodificador.
 - g. Escolha Próximo.

Note

- Você pode carregar apenas um arquivo .dbc para cada interface de rede.
- Verifique se cada sinal especificado no modelo do seu veículo está emparelhado com um sinal de decodificador.
- Depois de optar por adicionar outra interface de rede, você não poderá editar aquela que está editando. Você pode excluir as interfaces de rede existentes.

Etapa 3: revisar e criar

Verifique as configurações do manifesto do decodificador e selecione Criar.

Criar um manifesto do decodificador (AWS CLI)

Você pode usar a [CreateDecoderManifest](#) API operação para criar manifestos do decodificador. O exemplo a seguir usa a AWS CLI.

Important

Antes de criar um manifesto do decodificador, primeiro crie um modelo de veículo. Para obter mais informações, consulte [Crie um modelo de AWS veículo de IoT FleetWise](#).

Para criar um manifesto do decodificador, execute o seguinte comando.

Substituir *decoder-manifest-configuration* com o nome do JSON arquivo que contém a configuração.

```
aws iotfleetwise create-decoder-manifest --cli-input-json file://decoder-manifest-configuration.json
```

- Substituir *decoder-manifest-name* com o nome do manifesto do decodificador que você está criando.
- Substituir *vehicle-model-ARN* com o Amazon Resource Name (ARN) do modelo do veículo.
- (Opcional) Substituir *description* com uma descrição para ajudá-lo a identificar o manifesto do decodificador.

Para obter mais informações sobre como configurar ramificações, atributos, sensores e atuadores, consulte [Configurar interfaces de FleetWise rede AWS de IoT e sinais de decodificador](#).

```
{
  "name": "decoder-manifest-name",
  "modelManifestArn": "vehicle-model-arn",
  "description": "description",
  "networkInterfaces": [
    {
      "canInterface": {
        "name": "myNetworkInterface",
        "protocolName": "CAN",
        "protocolVersion": "2.0b"
      }
    }
  ]
}
```

```

    },
    "interfaceId": "Qq1acaenBy0B3sSM39SYm",
    "type": "CAN_INTERFACE"
  }
],
"signalDecoders": [
  {
    "canSignal": {
      "name": "Engine_Idle_Time",
      "factor": 1,
      "isBigEndian": true,
      "isSigned": false,
      "length": 24,
      "messageId": 271343712,
      "offset": 0,
      "startBit": 16
    },
    "fullyQualified_name": "Vehicle.EngineIdleTime",
    "interfaceId": "Qq1acaenBy0B3sSM39SYm",
    "type": "CAN_SIGNAL"
  },
  {
    "canSignal": {
      "name": "Engine_Run_Time",
      "factor": 1,
      "isBigEndian": true,
      "isSigned": false,
      "length": 24,
      "messageId": 271343712,
      "offset": 0,
      "startBit": 40
    },
    "fullyQualified_name": "Vehicle.EngineRunTime",
    "interfaceId": "Qq1acaenBy0B3sSM39SYm",
    "type": "CAN_SIGNAL"
  }
]
}

```

- Substituir *decoder-manifest-name* com o nome do manifesto do decodificador que você está criando.
- Substituir *vehicle-model-ARN* com o Amazon Resource Name (ARN) do modelo do veículo.

- (Opcional) Substituir *description* com uma descrição para ajudá-lo a identificar o manifesto do decodificador.

A ordem dos nós de propriedade dentro de uma estrutura (struct) deve permanecer consistente conforme definido no catálogo de sinais e no modelo do veículo (manifesto do modelo). Para obter mais informações sobre como configurar ramificações, atributos, sensores e atuadores, consulte [Configurar interfaces de FleetWise rede AWS de IoT e sinais de decodificador](#).

```
{
  "name": "decoder-manifest-name",
  "modelManifestArn": "vehicle-model-arn",
  "description": "description",
  "networkInterfaces": [{
    "canInterface": {
      "name": "myNetworkInterface",
      "protocolName": "CAN",
      "protocolVersion": "2.0b"
    },
    "interfaceId": "Qq1acaenBy0B3sSM39SYm",
    "type": "CAN_INTERFACE"
  }, {
    "type": "VEHICLE_MIDDLEWARE",
    "interfaceId": "G1KzxkdnmV5Hn7wkV3ZL9",
    "vehicleMiddleware": {
      "name": "ROS2_test",
      "protocolName": "ROS_2"
    }
  }
],
  "signalDecoders": [{
    "canSignal": {
      "name": "Engine_Idle_Time",
      "factor": 1,
      "isBigEndian": true,
      "isSigned": false,
      "length": 24,
      "messageId": 271343712,
      "offset": 0,
      "startBit": 16
    },
    "fullyQualifiedName": "Vehicle.EngineIdleTime",
    "interfaceId": "Qq1acaenBy0B3sSM39SYm",
    "type": "CAN_SIGNAL"
  }
]
```

```

},
{
  "canSignal": {
    "name": "Engine_Run_Time",
    "factor": 1,
    "isBigEndian": true,
    "isSigned": false,
    "length": 24,
    "messageId": 271343712,
    "offset": 0,
    "startBit": 40
  },
  "fullyQualifiedName": "Vehicle.EngineRunTime",
  "interfaceId": "Qq1lacaenBy0B3sSM39SYm",
  "type": "CAN_SIGNAL"
},
{
  "fullyQualifiedName": "Vehicle.CompressedImageTopic",
  "type": "MESSAGE_SIGNAL",
  "interfaceId": "G1KzxxkdnmV5Hn7wkV3ZL9",
  "messageSignal": {
    "topicName": "CompressedImageTopic:sensor_msgs/msg/CompressedImage",
    "structuredMessage": {
      "structuredMessageDefinition": [{
        "fieldName": "header",
        "dataType": {
          "structuredMessageDefinition": [{
            "fieldName": "stamp",
            "dataType": {
              "structuredMessageDefinition": [{
                "fieldName": "sec",
                "dataType": {
                  "primitiveMessageDefinition": {
                    "ros2PrimitiveMessageDefinition": {
                      "primitiveType": "INT32"
                    }
                  }
                }
              ]
            }
          ]
        },
        "fieldName": "nanosec",
        "dataType": {
          "primitiveMessageDefinition": {
            "ros2PrimitiveMessageDefinition": {

```

```
        "primitiveType": "UINT32"
      }
    }
  ]
}
},
{
  "fieldName": "frame_id",
  "dataType": {
    "primitiveMessageDefinition": {
      "ros2PrimitiveMessageDefinition": {
        "primitiveType": "STRING"
      }
    }
  }
}
]
}
},
{
  "fieldName": "format",
  "dataType": {
    "primitiveMessageDefinition": {
      "ros2PrimitiveMessageDefinition": {
        "primitiveType": "STRING"
      }
    }
  }
},
{
  "fieldName": "data",
  "dataType": {
    "structuredMessageListDefinition": {
      "name": "listType",
      "memberType": {
        "primitiveMessageDefinition": {
          "ros2PrimitiveMessageDefinition": {
            "primitiveType": "UINT8"
          }
        }
      }
    }
  },
  "capacity": 0,
```



```
        "listType": "DYNAMIC_UNBOUNDED_CAPACITY"
      }
    }
  ]
}
}
]
```

Note

Você pode baixar um [script de demonstração](#) para criar um manifesto do decodificador com sinais de sistemas de visão. Para ter mais informações, consulte o [Vision System Data Developer Guide](#).

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Atualizar um manifesto AWS do decodificador de IoT FleetWise

Você pode usar a [UpdateDecoderManifestAPI](#) operação para atualizar um manifesto do decodificador. Você pode adicionar, remover e atualizar interfaces de rede e decodificadores de sinal. Você também pode alterar o status do manifesto do decodificador. O exemplo a seguir usa a AWS CLI.

Para atualizar um manifesto do decodificador, execute o comando a seguir.

Substituir *decoder-manifest-name* com o nome do manifesto do decodificador que você está atualizando.

```
aws iotfleetwise update-decoder-manifest /
    --name decoder-manifest-name /
    --status ACTIVE
```

Important

Depois de ativar o manifesto do decodificador, você não poderá editá-lo.

Excluir um manifesto AWS do decodificador de IoT FleetWise

Você pode usar o FleetWise console de AWS IoT ou excluir um API manifesto do decodificador.

Important

Os veículos associados ao manifesto do decodificador devem ser excluídos primeiro. Para obter mais informações, consulte [Excluir um veículo AWS de IoT FleetWise](#).

Tópicos

- [Excluir um manifesto do decodificador \(console\)](#)
- [Excluir um manifesto do decodificador \(AWS CLI\)](#)

Excluir um manifesto do decodificador (console)

Você pode usar o FleetWise console de AWS IoT para excluir um manifesto do decodificador.

Para excluir um manifesto do decodificador

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Modelos de veículos.
3. Selecione o modelo de veículo de destino.
4. Na página de resumo do modelo do veículo, selecione a guia Manifestos do decodificador.
5. Selecione o manifesto do decodificador de destino e depois Excluir.
6. Em Excluir **decoder-manifest-name?**, insira o nome do manifesto do decodificador a ser excluído e selecione Confirmar.

Excluir um manifesto do decodificador (AWS CLI)

Você pode usar a [DeleteDecoderManifest](#) API operação para excluir um manifesto do decodificador. O exemplo a seguir usa AWS CLI.

⚠ Important

Antes de excluir o manifesto do decodificador, exclua primeiro os veículos associados. Para obter mais informações, consulte [Excluir um veículo AWS de IoT FleetWise](#).

Para excluir um manifesto do decodificador, execute o comando a seguir.

Substituir *decoder-manifest-name* com o nome do manifesto do decodificador que você está excluindo.

```
aws iotfleetwise delete-decoder-manifest --name decoder-manifest-name
```

Obtenha informações sobre o manifesto do FleetWise decodificador AWS de IoT

Você pode usar a [ListDecoderManifests](#) API operação para verificar se um manifesto do decodificador foi excluído. O exemplo a seguir usa AWS CLI.

Para recuperar uma lista paginada de resumos de todos os manifestos do decodificador, execute o comando a seguir.

```
aws iotfleetwise list-decoder-manifests
```

Você pode usar a [ListDecoderManifestSignals](#) API operação para verificar se os sinais do decodificador no manifesto do decodificador foram atualizados. O exemplo a seguir usa AWS CLI.

Para recuperar uma lista paginada de resumos de todos os sinais do decodificador (nós) em um determinado manifesto do decodificador, execute o comando a seguir.

Substituir *decoder-manifest-name* com o nome do manifesto do decodificador que você está verificando.

```
aws iotfleetwise list-decoder-manifest-signals /  
    --name decoder-manifest-name
```

Você pode usar a [ListDecoderManifestNetworkInterfaces](#) API operação para verificar se as interfaces de rede no manifesto do decodificador foram atualizadas. O exemplo a seguir usa AWS CLI.

Para recuperar uma lista paginada de resumos de todas as interfaces de rede em um determinado manifesto do decodificador, execute o comando a seguir.

Substituir *decoder-manifest-name* com o nome do manifesto do decodificador que você está verificando.

```
aws iotfleetwise list-decoder-manifest-network-interfaces /  
    --name decoder-manifest-name
```

Você pode usar a [GetDecoderManifest](#) API operação para verificar se as interfaces de rede e os sinais do decodificador no manifesto do decodificador foram atualizados. O exemplo a seguir usa AWS CLI.

Para recuperar informações sobre um manifesto do decodificador, execute o comando a seguir.

Substituir *decoder-manifest* com o nome do manifesto do decodificador que você deseja recuperar.

```
aws iotfleetwise get-decoder-manifest --name decoder-manifest
```

Note

Essa operação é [eventualmente consistente](#). Em outras palavras, as alterações no manifesto do decodificador podem não ser refletidas imediatamente.

Gerencie AWS veículos de IoT FleetWise

Veículos são instâncias de modelos de veículos. Os veículos devem ser criados com base em um modelo de veículo e associados a um manifesto do decodificador. Os veículos carregam um ou mais fluxos de dados para a nuvem. Por exemplo, um veículo pode enviar dados de quilometragem, temperatura do motor e estado do ar quente para a nuvem. Cada veículo contém as seguintes informações:

`vehicleName`

Um ID que identifica o veículo.

Não adicione informações de identificação pessoal (PII) ou outras informações confidenciais ou sensíveis no nome do seu veículo. Os nomes dos veículos podem ser acessados por outros AWS serviços, incluindo a Amazon CloudWatch. Os nomes dos veículos não se destinam a ser usados para dados privados ou confidenciais.

`modelManifestARN`

O nome de recurso da Amazon (ARN) de um modelo de veículo (manifesto do modelo). Cada veículo é criado usando um modelo de veículo. Os veículos criados usando o mesmo modelo de veículo consistem no mesmo grupo de sinais herdados do modelo de veículo. Esses sinais são definidos e padronizados no catálogo de sinais.

`decoderManifestArn`

O ARN do manifesto do decodificador. Um manifesto do decodificador fornece informações de decodificação que a AWS IoT FleetWise pode usar para transformar dados brutos do sinal (dados binários) em valores legíveis por humanos. Um manifesto do decodificador deve estar associado a um modelo de veículo. AWS A IoT FleetWise usa o mesmo manifesto do decodificador para decodificar dados brutos de veículos criados com base no mesmo modelo de veículo.

`attributes`

Atributos são pares de valores-chave que contêm informações estáticas. Os veículos podem conter atributos herdados do modelo do veículo. Você pode adicionar outros atributos para distinguir um veículo individual de outros veículos criados usando o mesmo modelo de veículo. Por exemplo, se você tiver um carro preto, poderá especificar o seguinte valor para um atributo: `{"color": "black"}`.

⚠ Important

Os atributos devem ser definidos no modelo de veículo associado antes de poder adicioná-los a veículos individuais.

Para obter mais informações sobre modelos de veículos, manifestos de decodificadores e atributos, consulte [Modelos de AWS veículos de IoT FleetWise](#).

AWS A IoT FleetWise fornece as seguintes API operações que você pode usar para criar e gerenciar veículos.

- [CreateVehicle](#)— Cria um novo veículo.
- [BatchCreateVehicle](#)— Cria um ou mais veículos novos.
- [UpdateVehicle](#)— Atualiza um veículo existente.
- [BatchUpdateVehicle](#)— Atualiza um ou mais veículos existentes.
- [DeleteVehicle](#)— Exclui um veículo existente.
- [ListVehicles](#)— Recupera uma lista paginada de resumos de todos os veículos.
- [GetVehicle](#)— Recupera informações sobre um veículo.

Tutoriais

- [Provisionar AWS veículos de IoT FleetWise](#)
- [Tópicos reservados em AWS IoT FleetWise](#)
- [Crie um veículo AWS de IoT FleetWise](#)
- [Crie vários veículos AWS de IoT FleetWise](#)
- [Atualizar um veículo AWS de IoT FleetWise](#)
- [Atualize vários veículos AWS de IoT FleetWise](#)
- [Excluir um veículo AWS de IoT FleetWise](#)
- [Obtenha informações sobre AWS veículos de IoT FleetWise](#)

Provisionar AWS veículos de IoT FleetWise

O FleetWise software Edge Agent para AWS IoT executado em seu veículo coleta e transfere dados para a nuvem. AWS A IoT se FleetWise integra AWS IoT Core para oferecer suporte à comunicação

segura entre o software Edge Agent e a nuvem por meio de MQTT. Cada veículo corresponde a uma coisa AWS IoT. Você pode usar algo existente para criar um veículo ou configurar a AWS IoT FleetWise para criar automaticamente algo para seu veículo. Para obter mais informações, consulte [Criar um veículo \(AWS CLI\)](#).

AWS IoT Core oferece suporte à [autenticação](#) e [autorização](#) que ajudam a controlar com segurança o acesso aos recursos de AWS IoT. Os veículos podem usar certificados X.509 para serem autenticados (conectados) para usar a AWS IoT FleetWise e políticas para serem autorizados (ter permissões) a realizar ações específicas.

Autenticar veículos

Você pode criar políticas para autenticar seus veículos.

Para autenticar seu veículo

- Para criar uma política, execute o comando a seguir.
 - Substituir *policy-name* com o nome da política que você deseja criar.
 - Substituir *file-name* com o nome do JSON arquivo que contém a política.

```
aws iot create-policy --policy-name policy-name --policy-document file://file-name.json
```

Antes de usar a política de exemplo, faça o seguinte:

- Substituir *region* com a AWS região em que você criou recursos de AWS IoT. FleetWise
- Substituir *awsAccount* com o ID AWS da sua conta.

Este exemplo inclui tópicos reservados pela AWS IoT FleetWise. Você deve adicionar os tópicos à política. Para obter mais informações, consulte [Tópicos reservados em AWS IoT FleetWise](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```

        "iot:Connect"
    ],
    "Resource": [
        "arn:aws:iot:region:awsAccount:client/
${iot:Connection.Thing.ThingName}"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "iot:Publish"
    ],
    "Resource": [
        "arn:aws:iot:region:awsAccount:topic/$aws/iotfleetwise/vehicles/
${iot:Connection.Thing.ThingName}/checkins",
        "arn:aws:iot:region:awsAccount:topic/$aws/iotfleetwise/vehicles/
${iot:Connection.Thing.ThingName}/signals"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "iot:Subscribe"
    ],
    "Resource": [
        "arn:aws:iot:region:awsAccount:topicfilter/$aws/iotfleetwise/
vehicles/${iot:Connection.Thing.ThingName}/collection_schemes",
        "arn:aws:iot:region:awsAccount:topicfilter/$aws/iotfleetwise/
vehicles/${iot:Connection.Thing.ThingName}/decoder_manifests"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "iot:Receive"
    ],
    "Resource": [
        "arn:aws:iot:region:awsAccount:topic/$aws/iotfleetwise/vehicles/
${iot:Connection.Thing.ThingName}/collection_schemes",
        "arn:aws:iot:region:awsAccount:topic/$aws/iotfleetwise/vehicles/
${iot:Connection.Thing.ThingName}/decoder_manifests"
    ]
}
]

```



```
}
```

Autorizar veículos

Você pode criar certificados X.509 para autorizar seus veículos.

Para autorizar seu veículo

Important

Recomendamos que você crie um novo certificado para cada veículo.

1. Para criar um RSA key pair e emitir um certificado X.509, execute o comando a seguir.
 - Substituir *cert* com o nome do arquivo que salva o conteúdo de saída do comandocertificatePem.
 - Substituir *public-key* com o nome do arquivo que salva o conteúdo de saída do comandokeyPair. PublicKey.
 - Substituir *private-key* com o nome do arquivo que salva o conteúdo de saída do comandokeyPair. PrivateKey.

```
aws iot create-keys-and-certificate \  
  --set-as-active \  
  --certificate-pem-outfile cert.pem \  
  --public-key-outfile public-key.key" \  
  --private-key-outfile private-key.key"
```

2. Copie o Amazon Resource Name (ARN) do certificado da saída.
3. Para anexar a política ao certificado, execute o comando a seguir.
 - Substituir *policy-name* com o nome da AWS IoT Core política que você criou.
 - Substituir *certificate-arn* com o ARN do certificado que você copiou.

```
aws iot attach-policy \  
  --policy-name policy-name
```

```
--target "certificate-arn"
```

4. Para anexar o certificado à coisa, execute o seguinte comando.

- Substituir *thing-name* com o nome da sua AWS IoT coisa ou a identificação do seu veículo.
- Substituir *certificate-arn* com o ARN do certificado que você copiou.

```
aws iot attach-thing-principal \  
  --thing-name thing-name \  
  --principal "certificate-arn"
```

Tópicos reservados em AWS IoT FleetWise

AWS A IoT FleetWise reserva o uso dos seguintes tópicos. Se o tópico reservado permitir, você poderá se inscrever ou publicar nele. No entanto, não é possível criar novos tópicos que comecem com um cifrão (\$). Se você usar operações de publicação ou assinatura incompatível com tópicos reservados, isso poderá resultar no fim da conexão.

Tópico	Operação do cliente permitida	Descrição
\$aws/iotfleetwise/vehicles/ <i>vehicleName</i> /checkins	Publicar	<p>O software Edge Agent publica informações de status do veículo neste tópico.</p> <p>As informações de status do veículo são trocadas em formato de buffers de protocolo (protobuf). Para obter mais informações, consulte o Guia do desenvolvedor de FleetWise</p>

Tópico	Operação do cliente permitida	Descrição
		software do Edge Agent para AWS IoT.
<code>\$aws/iotfleetwise/vehicles/<i>vehicleName</i> /signals</code>	Publicar	<p>O software Edge Agent publica sinais para este tópico.</p> <p>As informações de sinal são trocadas no formato de buffers de protocolo (protobuf). Para obter mais informações, consulte o Guia do desenvolvedor de FleetWise software do Edge Agent para AWS IoT.</p>
<code>\$aws/iotfleetwise/vehicles/<i>vehicleName</i> /collection_schemes</code>	Assinar	AWS A IoT FleetWise publica esquemas de coleta de dados sobre esse tópico. Os veículos consomem esses esquemas de coleta de dados.
<code>\$aws/iotfleetwise/vehicles/<i>vehicleName</i> /decoder_manifests</code>	Assinar	AWS A IoT FleetWise publica manifestos de decodificadores para esse tópico. Os veículos consomem esses manifestos do decodificador.

Crie um veículo AWS de IoT FleetWise

Você pode usar o FleetWise console de AWS IoT ou API criar um veículo.

Important

Antes de começar, verifique o seguinte:

- Você deve ter um modelo de veículo, e o status dele deve ser ACTIVE. Para obter mais informações, consulte [Gerencie modelos AWS de veículos de IoT FleetWise](#).
- O modelo do veículo deve estar associado a um manifesto do decodificador, e o status do manifesto do decodificador deve ser ACTIVE. Para obter mais informações, consulte [Gerencie manifestos AWS de decodificadores de IoT FleetWise](#).

Tópicos

- [Criar um veículo \(console\)](#)
- [Criar um veículo \(AWS CLI\)](#)

Criar um veículo (console)

Você pode usar o FleetWise console de AWS IoT para criar um veículo.

Important

Antes de começar, verifique o seguinte:

- Você deve ter um modelo de veículo, e o status dele deve ser ACTIVE. Para obter mais informações, consulte [Gerencie modelos AWS de veículos de IoT FleetWise](#).
- O modelo do veículo deve estar associado a um manifesto do decodificador, e o status do manifesto do decodificador deve ser ACTIVE. Para obter mais informações, consulte [Gerencie manifestos AWS de decodificadores de IoT FleetWise](#).

Para criar um veículo

1. Abra o console [AWS de IoT. FleetWise](#)

2. No painel de navegação, selecione Veículos.
3. Na página de resumo do veículo, selecione Criar veículo e execute as etapas a seguir.

Tópicos

- [Etapa 1: definir as propriedades do veículo](#)
- [Etapa 2: configurar o certificado do veículo](#)
- [Etapa 3: anexar políticas ao certificado](#)
- [Etapa 4: revisar e criar](#)

Etapa 1: definir as propriedades do veículo

Nesta etapa, você dá um nome para o veículo e o associa ao manifesto do modelo e ao manifesto do decodificador.

1. Insira um nome para o veículo.

Important

Um veículo corresponde a qualquer AWS IoT coisa. Se já existir alguma coisa com esse nome, selecione Associar o veículo a uma coisa de IoT para atualizar a coisa com o veículo. Ou escolha um nome de veículo diferente e a AWS IoT FleetWise criará automaticamente algo novo para o veículo.

2. Selecione um modelo de veículo (manifesto do modelo) na lista.
3. Selecione um manifesto de decodificador na lista. O manifesto do decodificador está associado ao modelo do veículo.
4. (Opcional) Para associar atributos do veículo, escolha Adicionar atributos. Se você ignorar esta etapa, deverá adicionar atributos após a criação do veículo antes de poder implementá-lo em campanhas.
5. (Opcional) Para associar etiquetas ao veículo, selecione Adicionar nova etiqueta. Você também poderá adicionar tags após a criação do veículo.
6. Escolha Próximo.

Etapa 2: configurar o certificado do veículo

Para usar seu veículo como uma AWS IoT coisa, você deve configurar um certificado de veículo com uma política anexada. Se você ignorar esta etapa, deverá configurar um certificado após a criação do veículo antes de poder implantá-lo em campanhas.

1. Selecione Gerar automaticamente um novo certificado (recomendado).
2. Escolha Próximo.

Etapa 3: anexar políticas ao certificado

Anexar uma política ao certificado que você configurou na etapa anterior.

1. Em Políticas, insira um nome de política existente. Para criar uma política, selecione Criar política.
2. Escolha Próximo.

Etapa 4: revisar e criar

Verifique as configurações do veículo e selecione Criar veículo.

Important

Depois que o veículo for criado, você deverá baixar o certificado e as chaves. Você usará o certificado e a chave privada para conectar o veículo no software Edge Agent for AWS IoT FleetWise .

Criar um veículo (AWS CLI)

Ao criar um veículo, você deve usar um modelo de veículo associado a um manifesto do decodificador. Você pode usar a [CreateVehicle](#) API operação para criar um veículo. O exemplo a seguir usa a AWS CLI.

Important

Antes de começar, verifique o seguinte:

- Você deve ter um modelo de veículo, e o status dele deve ser ACTIVE. Para obter mais informações, consulte [Gerencie modelos AWS de veículos de IoT FleetWise](#).
- O modelo do veículo deve estar associado a um manifesto do decodificador, e o status do manifesto do decodificador deve ser ACTIVE. Para obter mais informações, consulte [Gerencie manifestos AWS de decodificadores de IoT FleetWise](#).

Para criar um veículo, execute o seguinte comando.

Substituir *file-name* com o nome do JSON arquivo que contém a configuração do veículo.

```
aws iotfleetwise create-vehicle --cli-input-json file://file-name.json
```

Exemplo configuração do veículo

- (Opcional) O valor `associationBehavior` pode ser um dos seguintes:
 - `CreateIotThing`— Quando seu veículo é criado, AWS a IoT cria FleetWise automaticamente AWS IoT algo com o nome do ID do veículo.
 - `ValidateIotThingExists` — Use uma coisa da AWS IoT existente para criar um veículo.

Para criar AWS IoT algo, execute o comando a seguir. Substituir *thing-name* com o nome da coisa que você deseja criar.

```
aws iot create-thing --thing-name thing-name
```

Se não for especificado, a AWS IoT cria FleetWise automaticamente AWS IoT algo para o seu veículo.

Important

Certifique-se de que a AWS IoT coisa seja provisionada após a criação do veículo. Para obter mais informações, consulte [Provisionar AWS veículos de IoT FleetWise](#).

- Substituir *vehicle-name* com uma das seguintes opções.
 - O nome da sua AWS IoT coisa, se `associationBehavior` estiver configurado para `validateIotThingExists`.

- O ID do veículo a ser criado se `associationBehavior` estiver configurado como `CreateIotThing`.

O ID do veículo pode ter de 1 a 100 caracteres. Caracteres válidos: a–z, A–Z, 0–9, traço (-), sublinhado (_) e dois pontos (:).

- Substituir *model-manifest-ARN* com o modelo ARN do seu veículo (manifesto do modelo).
- Substituir *decoder-manifest-ARN* com o ARN do manifesto do decodificador associado ao modelo de veículo especificado.
- (Opcional) Você pode adicionar outros atributos para distinguir este veículo de outros veículos criados usando o mesmo modelo de veículo. Por exemplo, se você tiver um carro elétrico, poderá especificar o seguinte valor para um atributo: `{"fuelType": "electric"}`.

Important

Os atributos devem ser definidos no modelo de veículo associado antes de poder adicioná-los a veículos individuais.

```
{
  "associationBehavior": "associationBehavior",
  "vehicleName": "vehicle-name",
  "modelManifestArn": "model-manifest-ARN",
  "decoderManifestArn": "decoder-manifest-ARN",
  "attributes": {
    "key": "value"
  }
}
```

Crie vários veículos AWS de IoT FleetWise

Você pode usar a [BatchCreateVehicle](#) API operação para criar vários veículos ao mesmo tempo. O exemplo a seguir usa a AWS CLI.

Para criar vários veículos, execute o seguinte comando.

Substituir *file-name* com o nome do JSON arquivo que contém as configurações de vários veículos.


```
aws iotfleetwise batch-create-vehicle --cli-input-json file://file-name.json
```

Exemplo configurações de veículos

```
{
  "vehicles": [
    {
      "associationBehavior": "associationBehavior",
      "vehicleName": "vehicle-name",
      "modelManifestArn": "model-manifest-ARN",
      "decoderManifestArn": "decoder-manifest-ARN",
      "attributes": {
        "key": "value"
      }
    },
    {
      "associationBehavior": "associationBehavior",
      "vehicleName": "vehicle-name",
      "modelManifestArn": "model-manifest-ARN",
      "decoderManifestArn": "decoder-manifest-ARN",
      "attributes": {
        "key": "value"
      }
    }
  ]
}
```

Você pode criar até 10 veículos para cada operação em lote. Para obter mais informações sobre a configuração do veículo, consulte [Criar um veículo \(AWS CLI\)](#).

Atualizar um veículo AWS de IoT FleetWise

Você pode usar a [UpdateVehicle](#) API operação para atualizar um veículo existente. O exemplo a seguir usa a AWS CLI.

Para atualizar um veículo, execute o seguinte comando.

Substituir *file-name* com o nome do JSON arquivo que contém a configuração do seu veículo.

```
aws iotfleetwise update-vehicle --cli-input-json file://file-name.json
```

Exemplo configuração do veículo

- Substituir *vehicle-name* com a ID do veículo que você deseja atualizar.
- (Opcional) Substituir *model-manifest-ARN* com o modelo ARN do veículo (manifesto do modelo) que você usa para substituir o modelo do veículo em uso.
- (Opcional) Substituir *decoder-manifest-ARN* com o manifesto ARN do decodificador associado ao novo modelo de veículo que você especificou.
- (Opcional) Substituir *attribute-update-mode* com atributos do veículo.
 - Merge — Mesclar novos atributos em atributos existentes atualizando este último com novos valores e adicionando novos atributos se eles não existirem.

Por exemplo, se um veículo tiver os seguintes atributos: {"color": "black", "fuelType": "electric"}, e você atualizar o veículo com os seguintes atributos: {"color": "", "fuelType": "gasoline", "model": "x"}, o veículo atualizado terá os seguintes atributos: {"fuelType": "gasoline", "model": "x"}.

- Overwrite — Substitua os atributos existentes por novos.

Por exemplo, se um veículo tiver os seguintes atributos: {"color": "black", "fuelType": "electric"} e você atualizar o veículo pelo atributo {"model": "x"}, o veículo atualizado terá o atributo {"model": "x"}.

Isso será necessário se houver atributos na entrada.

- (Opcional) Para adicionar novos atributos ou atualizar os existentes com novos valores, configure `attributes`. Por exemplo, se você tiver um carro elétrico, poderá especificar o seguinte valor para um atributo: {"fuelType": "electric"}.

Para excluir atributos, configure `attributeUpdateMode` como Merge.

Important

Os atributos devem ser definidos no modelo de veículo associado antes de poder adicioná-los a veículos individuais.

```
{  
  "vehicleName": "vehicle-name",  
  "modelManifestArn": "model-manifest-arn",
```

```
    "decoderManifestArn": "decoder-manifest-arn",
    "attributeUpdateMode": "attribute-update-mode"
  }
}
```

Atualize vários veículos AWS de IoT FleetWise

Você pode usar a [BatchUpdateVehicle](#) API operação para atualizar vários veículos existentes ao mesmo tempo. O exemplo a seguir usa a AWS CLI.

Para atualizar vários veículos, execute o seguinte comando.

Substituir *file-name* com o nome do JSON arquivo que contém as configurações de vários veículos.

```
aws iotfleetwise batch-update-vehicle --cli-input-json file://file-name.json
```

Exemplo configurações de veículos

```
{
  "vehicles": [
    {
      "vehicleName": "vehicle-name",
      "modelManifestArn": "model-manifest-arn",
      "decoderManifestArn": "decoder-manifest-arn",
      "mergeAttributes": true,
      "attributes": {
        "key": "value"
      }
    },
    {
      "vehicleName": "vehicle-name",
      "modelManifestArn": "model-manifest-arn",
      "decoderManifestArn": "decoder-manifest-arn",
      "mergeAttributes": true,
      "attributes": {
        "key": "value"
      }
    }
  ]
}
```

Você pode atualizar até 10 veículos para cada operação em lote. Para obter mais informações sobre a configuração de cada veículo, consulte [Atualizar um veículo AWS de IoT FleetWise](#).

Excluir um veículo AWS de IoT FleetWise

Você pode usar o FleetWise console de AWS IoT ou API excluir veículos.

Important

Depois que um veículo é excluído, a AWS IoT remove FleetWise automaticamente o veículo das frotas e campanhas associadas. Para ter mais informações, consulte [Gerencie frotas na IoT AWS FleetWise](#) e [Colete FleetWise dados de AWS IoT com campanhas](#). No entanto, o veículo ainda existe como uma coisa ou ainda está associado a uma coisa dentro AWS IoT Core. Para obter instruções sobre como excluir uma coisa, consulte [Excluir uma coisa](#) no Guia do desenvolvedor do AWS IoT Core.

Excluir um veículo (console)

Você pode usar o FleetWise console de AWS IoT para excluir um veículo.

Para excluir um veículo

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Veículos.
3. Na página Veículos, selecione o botão ao lado do veículo que você quer excluir.
4. Escolha Excluir.
5. Em Excluir **vehicle-name**, insira o nome do veículo e selecione Excluir.

Excluir um veículo (AWS CLI)

Você pode usar a [DeleteVehicle](#) API operação para excluir um veículo. O exemplo a seguir usa AWS CLI.

Para excluir um veículo, execute o seguinte comando.

Substituir *vehicle-name* com a ID do veículo que você deseja excluir.

```
aws iotfleetwise delete-vehicle --vehicle-name vehicle-name
```

Obtenha informações sobre AWS veículos de IoT FleetWise

Você pode usar a [ListVehicles](#) API operação para verificar se um veículo foi excluído. O exemplo a seguir usa a AWS CLI.

Para recuperar uma lista paginada de resumos de todos os veículos, execute o seguinte comando.

```
aws iotfleetwise list-vehicles
```

Você pode usar a [GetVehicle](#) API operação para recuperar as informações do veículo. O exemplo a seguir usa a AWS CLI.

Para recuperar os metadados de um veículo, execute o seguinte comando.

Substituir *vehicle-name* com o ID do veículo que você deseja recuperar.

```
aws iotfleetwise get-vehicle --vehicle-name vehicle-name
```

Note

Essa operação é [eventualmente consistente](#). Em outras palavras, as alterações no veículo podem não ser refletidas imediatamente.

Gerencie frotas na IoT AWS FleetWise

Uma frota representa um grupo de veículos. Uma frota sem veículos associados é uma entidade vazia. Antes de poder usar a frota para gerenciar vários veículos ao mesmo tempo, você deve associar os veículos à frota. Um veículo pode pertencer a várias frotas. Você pode controlar quais dados coletar de uma frota de veículos e quando coletar dados implantando uma campanha. Para obter mais informações, consulte [Colete FleetWise dados de AWS IoT com campanhas](#).

Uma frota contém as seguintes informações.

`fleetId`

O ID da frota.

(Opcional) `description`

Uma descrição que ajuda você a encontrar a frota.

`signalCatalogArn`

O Amazon Resource Name (ARN) do catálogo de sinais.

AWS A IoT FleetWise fornece as seguintes API operações que você pode usar para criar e gerenciar frotas.

- [CreateFleet](#)— Cria um grupo de veículos que contém o mesmo grupo de sinais.
- [AssociateVehicleFleet](#)— Associa um veículo a uma frota.
- [DisassociateVehicleFleet](#)— Desassocia um veículo de uma frota.
- [UpdateFleet](#)— Atualiza a descrição de uma frota existente.
- [DeleteFleet](#)— Exclui uma frota existente.
- [ListFleets](#)— Recupera uma lista paginada de resumos de todas as frotas.
- [ListFleetsForVehicle](#)— Recupera uma lista paginada IDs de todas as frotas às quais o veículo pertence.
- [ListVehiclesInFleet](#)— Recupera uma lista paginada de resumos de todos os veículos em uma frota.
- [GetFleet](#)— Recupera informações sobre uma frota.

Tópicos

- [Crie uma frota AWS de IoT FleetWise](#)
- [Associe um FleetWise veículo de AWS IoT a uma frota](#)
- [Desassociar um veículo AWS de FleetWise IoT de uma frota](#)
- [Atualize uma frota AWS de IoT FleetWise](#)
- [Excluir uma frota AWS de IoT FleetWise](#)
- [Obtenha informações sobre AWS a frota de IoT FleetWise](#)

Crie uma frota AWS de IoT FleetWise

Você pode usar a [CreateFleet](#) API operação para criar uma frota de veículos. O exemplo a seguir usa AWS CLI.

Important

Você deve ter um catálogo de sinais antes de poder criar uma frota. Para obter mais informações, consulte [Crie um catálogo de AWS sinais de IoT FleetWise](#).

Para criar uma frota, execute o seguinte comando.

- Substituir *fleet-id* com o ID da frota que você está criando.

O ID da frota deve ser exclusivo e ter de 1 a 100 caracteres. Caracteres válidos: letras (A–Z e a–z), números (0–9), dois pontos (:), hífen (-) e sublinhados (_).

- (Opcional) Substituir *description* com uma descrição.

A descrição pode conter de 1 a 2.048 caracteres.

- Substituir *signal-catalog-arn* com o ARN do catálogo de sinais.

```
aws iotfleetwise create-fleet \  
  --fleet-id fleet-id \  
  --description description \  
  --signal-catalog-arn signal-catalog-arn
```

Associe um FleetWise veículo de AWS IoT a uma frota

Você pode usar a [AssociateVehicleFleet](#) API operação para associar um veículo a uma frota. O exemplo a seguir usa AWS CLI.

Important

- Você deve ter um veículo e uma frota antes de poder associar um veículo a uma frota. Para obter mais informações, consulte [Gerencie AWS veículos de IoT FleetWise](#).
- Se você associar um veículo a uma frota que é alvo de uma campanha, a AWS IoT implanta FleetWise automaticamente a campanha no veículo.

Para associar um veículo a uma frota, execute o seguinte comando.

- Substituir *fleet-id* com o ID da frota.
- Substituir *vehicle-name* com a identificação do veículo.

```
aws iotfleetwise associate-vehicle-fleet --fleet-id fleet-id --vehicle-name vehicle-name
```

Desassociar um veículo AWS de FleetWise IoT de uma frota

Você pode usar a [DisassociateVehicleFleet](#) API operação para desassociar um veículo de uma frota. O exemplo a seguir usa AWS CLI.

Para desassociar um veículo de uma frota, execute o seguinte comando.

- Substituir *fleet-id* com o ID da frota.
- Substituir *vehicle-name* com a identificação do veículo.

```
aws iotfleetwise disassociate-vehicle-fleet --fleet-id fleet-id --vehicle-name vehicle-name
```


Atualize uma frota AWS de IoT FleetWise

Você pode usar a [UpdateFleet](#) API operação para atualizar a descrição de uma frota. O exemplo a seguir usa AWS CLI.

Para atualizar uma frota, execute o comando a seguir.

- Substituir *fleet-id* com o ID da frota que você está atualizando.
- Substituir *description* com uma nova descrição.

A descrição pode conter de 1 a 2.048 caracteres.

```
aws iotfleetwise update-fleet --fleet-id fleet-id --description description
```

Excluir uma frota AWS de IoT FleetWise

Você pode usar a [DeleteFleet](#) API operação para excluir uma frota. O exemplo a seguir usa AWS CLI.

Important

Antes de excluir uma frota, confirme se ela não tem veículos associados. Para obter instruções sobre como desassociar um veículo de uma frota, consulte [Desassociar um veículo AWS de FleetWise IoT de uma frota](#).

Para excluir uma frota, execute o comando a seguir.

Substituir *fleet-id* com o ID da frota que você está excluindo.

```
aws iotfleetwise delete-fleet --fleet-id fleet-id
```

Obtenha informações sobre AWS a frota de IoT FleetWise

Você pode usar a [ListFleets](#) API operação para verificar se uma frota foi excluída. O exemplo a seguir usa a AWS CLI.

Para recuperar uma lista paginada de resumos de todas as frotas, execute o comando a seguir.

```
aws iotfleetwise list-fleets
```

Você pode usar a [ListFleetsForVehicle](#) API operação para recuperar uma lista paginada IDs de todas as frotas às quais o veículo pertence. O exemplo a seguir usa a AWS CLI.

Para recuperar uma lista paginada IDs de todas as frotas às quais o veículo pertence, execute o comando a seguir.

Substituir *vehicle-name* com a identificação do veículo.

```
aws iotfleetwise list-fleets-for-vehicle \  
  --vehicle-name vehicle-name
```

Você pode usar a [ListVehiclesInFleet](#) API operação para recuperar uma lista paginada de resumos de todos os veículos em uma frota. O exemplo a seguir usa a AWS CLI.

Para recuperar uma lista paginada de resumos de todos os veículos de uma frota, execute o seguinte comando.

Substituir *fleet-id* com o ID da frota.

```
aws iotfleetwise list-vehicles-in-fleet \  
  --fleet-id fleet-id
```

Você pode usar a [GetFleet](#) API operação para recuperar informações da frota. O exemplo a seguir usa a AWS CLI.

Para recuperar os metadados de uma frota, execute o comando a seguir.

Substituir *fleet-id* com o ID da frota.

```
aws iotfleetwise get-fleet \  
  --fleet-id fleet-id
```

Note

Essa operação é [eventualmente consistente](#). Em outras palavras, as alterações na frota podem não ser refletidas imediatamente.

Colete FleetWise dados de AWS IoT com campanhas

Uma campanha é uma orquestração de regras de coleta de dados. As campanhas fornecem instruções ao FleetWise software Edge Agent para AWS IoT sobre como selecionar, coletar e transferir dados para a nuvem.

Você cria as campanhas na nuvem. Depois que você ou sua equipe aprovam uma campanha, a AWS IoT a implanta FleetWise automaticamente nos veículos. Você pode escolher implantar uma campanha em um veículo ou em uma frota de veículos. O software Edge Agent não inicia a coleta de dados até que uma campanha em execução seja implantada no veículo.

Note

As campanhas não funcionarão até que você tenha o seguinte.

- O software Edge Agent está em execução no seu veículo. Para obter mais informações sobre como desenvolver, instalar e trabalhar com o software Edge Agent, faça o seguinte.
 1. Navegue até o console [AWS de IoT. FleetWise](#)
 2. Na página inicial do serviço, na FleetWise seção Introdução à AWS IoT, escolha Explore o Edge Agent.
- Você se configurou AWS IoT Core para provisionar seu veículo. Para obter mais informações, consulte [Provisionar AWS veículos de IoT FleetWise](#).

Cada item da lista contém as seguintes informações.

`signalCatalogArn`

O Amazon Resource Name (ARN) do catálogo de sinais associado à campanha.

(Opcional) `tags`

Tags são metadados que podem ser usados para gerenciar a campanha. Você pode atribuir a mesma tag a recursos de serviços diferentes para indicar que os recursos estão relacionados.

`TargetArn`

O ARN de um veículo ou frota no qual a campanha é implantada.

name

Um nome exclusivo que ajuda a identificar a campanha.

collectionScheme

Os esquemas de coleta de dados fornecem instruções ao software Edge Agent sobre quais dados coletar ou quando coletá-los. AWS FleetWise Atualmente, a IoT suporta o esquema de coleta baseado em condições e o esquema de coleta baseado em tempo.

conditionBasedCollectionScheme

O esquema de coleta baseado em condições utiliza uma expressão lógica para reconhecer quais dados coletar. O software Edge Agent coleta dados quando a condição é atendida.

expression

A expressão lógica usada para reconhecer quais dados devem ser coletados. Por exemplo, se a expressão `$variable.`myVehicle.InVehicleTemperature` > 50.0` for especificada, o software Edge Agent coletará valores de temperatura superiores a 50,0. Consulte instruções de como escrever expressões em [Expressões lógicas para campanhas AWS de IoT FleetWise](#).

(Opcional) `triggerMode` pode ser um dos valores a seguir.

- `RISING_EDGE` — O software Edge Agent coleta dados somente quando a condição é atendida pela primeira vez. Por exemplo, `$variable.`myVehicle.AirBagDeployed` == true`.
- `ALWAYS` — O software Edge Agent coleta dados sempre que a condição é atendida.

(Opcional) `minimumTriggerIntervalMs`

A duração mínima entre dois eventos de coleta de dados, em milissegundos. Se um sinal mudar com frequência, você poderá coletar dados em uma taxa mais lenta.

(Opcional) `conditionLanguageVersion`

A versão da linguagem de expressão condicional.

timeBasedCollectionScheme

Ao definir um esquema de coleta baseado em tempo, especifique um período de tempo em milissegundos. O software Edge Agent usa o período de tempo para decidir com que frequência coletar dados. Por exemplo, se o período for de 120.000 milissegundos, o software Edge Agent coletará dados uma vez a cada dois minutos.

(Opcional) `compression`

Para economizar largura de banda sem fio e reduzir o tráfego de rede, você pode especificar [SNAPPY](#) compactação de dados em veículos.

Por padrão (OFF), o software Edge Agent não compacta dados.

`dataDestinationConfigs`

Selecione o destino para onde a campanha transferirá os dados do veículo. Você pode optar por armazenar dados no Amazon S3 ou no Amazon Timestream.

S3 é um mecanismo de armazenamento de dados econômico que oferece recursos duráveis de gerenciamento de dados e serviços de dados downstream. Você pode usar o S3 para dados relacionados ao comportamento de direção ou análise de manutenção de longo prazo.

Timestream é um mecanismo de persistência de dados que ajuda a identificar tendências e padrões quase em tempo real. Você pode usar o Timestream para dados de séries temporais, por exemplo, para analisar tendências passadas na velocidade ou frenagem do veículo.

(Opcional) `dataExtraDimensions`

Você pode adicionar um ou mais atributos para fornecer informações adicionais para um sinal.

(Opcional) `description`

Você pode adicionar uma descrição para ajudar a identificar o propósito da campanha.

(Opcional) `diagnosticsMode`

Quando o modo de diagnóstico está configurado para `SEND_ACTIVE_DTCS`, a campanha envia códigos de problemas de diagnóstico padrão armazenados (DTCs) que ajudam a identificar o que está errado com seu veículo. Por exemplo, P0097 indica que o módulo de controle do motor (ECM) determinou que a entrada do sensor de temperatura do ar de admissão 2 (IAT2) está abaixo da faixa normal do sensor.

Por padrão (OFF), o software Edge Agent não envia códigos de diagnóstico.

(Opcional) `expiryTime`

Você pode definir a data de expiração da sua campanha. Quando a campanha expira, o software Edge Agent para de coletar dados conforme especificado na campanha. Se diversas campanhas forem implantadas no veículo, o software Edge Agent usará outras campanhas para coletar dados.

Valor padrão: 253402243200 (31 de dezembro de 9999, 00:00:00UTC)

(Opcional) `postTriggerCollectionDuration`

Você pode definir uma duração de coleta pós-acionamento, para que o software Edge Agent continue coletando dados por um período especificado após um esquema ser invocado. Por exemplo, se um esquema de coleta baseado em condições com a seguinte expressão for invocado: `$variable.`myVehicle.Engine.RPM` > 7000.0`, o software Edge Agent continuará coletando valores de rotações por minuto (RPM) para o mecanismo. Mesmo que RPM só suba mais de 7000 uma vez, isso pode indicar que há um problema mecânico. Nesse caso, o software Edge Agent pode continuar coletando dados para ajudar a monitorar a condição.

Valor padrão: 0


(Opcional) `priority`

Você pode especificar um inteiro para indicar o nível de prioridade da campanha. Campanhas com um número menor têm prioridade mais alta. Se você implantar diversas campanhas em um veículo, as campanhas com maior prioridade serão iniciadas primeiro.

Valor padrão: 0

(Opcional) `signalsToCollect`

Uma lista de sinais dos quais os dados são coletados quando o esquema de coleta de dados é invocado.

 Important

Os sinais utilizados na expressão para o esquema de coleta baseado em condições devem ser especificados neste campo.

`name`

O nome do sinal do qual os dados são coletados quando o esquema de coleta de dados é invocado.

(Opcional) `maxSampleCount`

O número máximo de amostras de dados que o software Edge Agent coleta e transfere para a nuvem quando o esquema de coleta de dados é chamado.

(Opcional) `minimumSamplingIntervalMs`

A duração mínima entre dois eventos de coleta de amostras de dados, em milissegundos. Se um sinal muda frequentemente, você pode usar este parâmetro para coletar dados em uma taxa mais lenta.

Intervalo válido: de 0 a 4294967295

(Opcional) `spoolingMode`

Se `spoolingMode` estiver configurado para `T0_DISK`, o software Edge Agent armazenará temporariamente os dados localmente quando um veículo não estiver conectado à nuvem. Após o restabelecimento da conexão, os dados armazenados localmente serão transferidos automaticamente para a nuvem.

Valor padrão: `OFF`

(Opcional) `startTime`

Uma campanha aprovada é ativada na hora de início.

Valor padrão: `0`

O status de uma campanha pode ser um dos valores a seguir.

- `CREATING`— AWS A IoT FleetWise está processando sua solicitação para criar a campanha.
- `WAITING_FOR_APPROVAL` — Depois que uma campanha é criada, ela entra no estado `WAITING_FOR_APPROVAL`. Para aprovar a campanha, use a `UpdateCampaign` API operação. Depois que a campanha é aprovada, a AWS IoT implanta FleetWise automaticamente a campanha no veículo ou frota alvo. Para obter mais informações, consulte [Atualizar uma campanha AWS de IoT FleetWise](#).
- `RUNNING` — A campanha está ativa.
- `SUSPENDED` — A campanha está suspensa. Para retomar a campanha, use a `UpdateCampaign` API operação.

AWS A IoT FleetWise fornece as seguintes API operações que você pode usar para criar e gerenciar campanhas.

- [CreateCampaign](#)— Cria uma nova campanha.

- [UpdateCampaign](#)— Atualiza uma campanha existente. Depois que uma campanha é criada, você deve usar essa API operação para aprová-la.
- [DeleteCampaign](#)— Exclui uma campanha existente.
- [ListCampaigns](#)— Recupera uma lista paginada de resumos de todas as campanhas.
- [GetCampaign](#)— Recupera informações sobre uma campanha.

Tutoriais

- [Crie uma campanha AWS de IoT FleetWise](#)
- [Atualizar uma campanha AWS de IoT FleetWise](#)
- [Excluir uma campanha AWS de IoT FleetWise](#)
- [Obtenha informações AWS da campanha de IoT FleetWise](#)

Crie uma campanha AWS de IoT FleetWise

Você pode usar o FleetWise console de AWS IoT ou criar campanhas API para coletar dados do veículo.

Important

Para que sua campanha funcione, você deve ter o seguinte:

- O software Edge Agent está em execução no seu veículo. Para obter mais informações sobre como desenvolver, instalar e trabalhar com o software Edge Agent, faça o seguinte:
 1. Navegue até o console [AWS de IoT. FleetWise](#)
 2. Na página inicial do serviço, na FleetWise seção Introdução à AWS IoT, escolha Explore o Edge Agent.
- Você se configurou AWS IoT Core para provisionar seu veículo. Para obter mais informações, consulte [Provisionar AWS veículos de IoT FleetWise](#).

Tópicos

- [Criar uma campanha \(console\)](#)
- [Criar uma campanha \(AWS CLI\)](#)
- [Expressões lógicas para campanhas AWS de IoT FleetWise](#)

Criar uma campanha (console)

Você pode usar o FleetWise console de AWS IoT para criar uma campanha para selecionar, coletar e transferir dados do veículo para a nuvem.

Como criar uma campanha

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Campanhas.
3. Na página Campanhas, selecione Criar campanha e, em seguida, conclua as etapas nos tópicos a seguir.

Tópicos

- [Etapa 1: configurar campanha](#)
- [Etapa 2: definir o destino do armazenamento](#)
- [Etapa 3: adicionar veículos](#)
- [Etapa 4: revisar e criar](#)
- [Etapa 5: implantar uma campanha](#)

Important

- Você deve ter um catálogo de sinalização e um veículo antes de criar uma campanha. Para ter mais informações, consulte [Gerencie AWS catálogos de sinais de IoT FleetWise](#) e [Gerencie AWS veículos de IoT FleetWise](#).
- Após a criação de uma campanha, você deverá aprová-la. Para obter mais informações, consulte [Etapa 5: implantar uma campanha](#).

Etapa 1: configurar campanha

Em Informações gerais, faça o seguinte:

1. Insira um nome para a campanha.
2. (Opcional) Insira uma descrição.

Configure o esquema de coleta de dados da campanha. Um esquema de coleta de dados fornece instruções ao software Edge Agent sobre quais dados coletar ou quando coletá-los. No FleetWise console de AWS IoT, você pode configurar um esquema de coleta de dados das seguintes formas:


- Defina manualmente o esquema de coleta de dados.
- Faça upload de um arquivo para definir automaticamente o esquema de coleta de dados.

Em Opção de configuração, selecione uma das seguintes opções:

- Para especificar manualmente o tipo de esquema de coleta de dados e definir opções para personalizar o esquema, selecione Definir esquema de coleta de dados.

Especifique manualmente o tipo de esquema de coleta de dados e defina opções para personalizar o esquema.

1. Na seção Detalhes do esquema de coleta de dados, selecione o tipo de esquema de coleta de dados para usar nesta campanha. Para usar uma expressão lógica para reconhecer quais dados do veículo coletar, selecione Baseado em condição. Para usar um período específico para decidir com que frequência coletar dados do veículo, selecione Baseado em tempo.
2. Defina por quanto tempo a campanha coletará dados.

 Note

Por padrão, uma campanha aprovada é ativada imediatamente e não tem horário de término definido. Para evitar cobranças extras, você deve especificar um intervalo de tempo.

3. Se você especificou um esquema de coleta de dados baseado em condições, deverá definir uma expressão lógica para reconhecer quais dados coletar. AWS IoT FleetWise usa uma expressão lógica para reconhecer quais dados coletar para um esquema baseado em condições. A expressão deve especificar o nome completo de um sinal como uma variável, um operador de comparação e um valor de comparação.


Por exemplo, se você especificar a expressão `$variable.`myVehicle.InVehicleTemperature` > 50.0`, a AWS IoT FleetWise coletará valores de temperatura maiores que 50,0. Consulte instruções de como escrever expressões em [Expressões lógicas para campanhas AWS de IoT FleetWise](#).

Insira a expressão lógica usada para reconhecer quais dados devem ser coletados.

4. (Opcional) Você pode especificar a versão do idioma da expressão condicional. O valor padrão é 1.
 5. (Opcional) Você pode especificar o intervalo mínimo de acionamento, que é a menor duração entre dois eventos de coleta de dados. Por exemplo, se um sinal muda frequentemente, você pode coletar dados em uma taxa mais lenta.
 6. Especifique a condição do Modo de acionamento para que o software Edge Agent colete dados. Por padrão, o FleetWise software Edge Agent para AWS IoT sempre coleta dados sempre que a condição é atendida. Ou poderá coletar dados somente quando a condição for atendida pela primeira vez, no primeiro acionamento.
 7. Se você tiver especificado um esquema de coleta de dados baseado em tempo, deverá especificar um Período, em milissegundos, de 10.000 a 60.000 milissegundos. O software Edge Agent usa o período de tempo para decidir com que frequência coletar dados.
 8. (Opcional) Você pode editar as Opções avançadas do esquema.
 - a. Para economizar largura de banda sem fio e reduzir o tráfego de rede compactando dados, selecione Snappy.
 - b. (Opcional) Para definir por quanto tempo, em milissegundos, continuará coletando dados após um evento de coleta de dados, você pode especificar a Duração da coleta pós-acionamento.
 - c. (Opcional) Para indicar o nível de prioridade da campanha, você pode especificar a Prioridade da campanha. As campanhas com um número menor de prioridade são implantadas primeiro e são consideradas de prioridade mais alta.
 - d. O software Edge Agent pode armazenar dados localmente temporariamente quando um veículo não está conectado à nuvem. Após o restabelecimento da conexão, os dados armazenados localmente serão transferidos automaticamente para a nuvem. Especifique se você quer que o Edge Agent armazene dados localmente durante uma conexão perdida.
 - e. (Opcional) Para fornecer informações adicionais para um sinal, adicione até cinco atributos como dimensões de dados extras.
- Para fazer upload de um arquivo para definir o esquema de coleta de dados, selecione Carregar um arquivo.json do seu dispositivo local. AWS IoT define FleetWise automaticamente quais opções você pode definir no arquivo. Você pode revisar e atualizar as opções selecionadas.

Faça upload de um arquivo .json com detalhes sobre o esquema de coleta de dados.

1. Para importar informações sobre o esquema de coleta de dados, selecione Escolher arquivos. Para obter mais informações sobre o formato de arquivo necessário, consulte a [CreateCampaignAPI documentação](#).


 Note

AWS FleetWise Atualmente, a IoT suporta a extensão de formato de arquivo.json.

2. AWS A IoT define FleetWise automaticamente o esquema de coleta de dados com base nas informações do seu arquivo. Analise as opções que a AWS IoT FleetWise selecionou para você. É possível atualizar as opções, se necessário.

Especifique os sinais


Você pode especificar os sinais dos quais coletar dados quando o esquema de coleta de dados for invocado.

 Important

Os sinais utilizados na expressão para o esquema de coleta baseado em condições devem ser especificados neste campo.

Para especificar os sinais dos quais coletar dados

1. Pesquise o nome totalmente qualificado do sinal.

 Note

O nome totalmente qualificado do sinal é o caminho para o sinal mais o nome do sinal. Use um ponto (.) para se referir ao sinal secundário.

Por exemplo,

`Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState`

é o nome totalmente qualificado do atuador `HandsOffSteeringState`.

`Vehicle.Chassis.SteeringWheel.HandsOff.` é o caminho para esse atuador.

2. (Opcional) Em Contagem máxima de amostras, insira o número máximo de amostras de dados que o software Edge Agent coleta e transfere para a nuvem quando o esquema de coleta de dados é invocado.
3. (Opcional) Em Intervalo mínimo de amostragem, insira a duração mínima de tempo entre dois eventos de coleta de amostra de dados, em milissegundos. Se um sinal muda frequentemente, você pode usar este parâmetro para coletar dados em uma taxa mais lenta.
4. Para adicionar outro sinal, selecione Adicionar mais sinais. Você pode adicionar até 999 sinais.
5. Escolha Próximo.

Etapa 2: definir o destino do armazenamento

Note

Você só poderá transferir dados do veículo para o Amazon S3 se a campanha contiver sinais de dados de sistemas de visão.

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Escolha o destino onde deseja armazenar os dados coletados pela campanha. É possível transferir dados de veículos para o Amazon S3 ou o Amazon Timestream.

Em Configurações de destino, faça o seguinte:

- Selecione S3 ou Timestream na lista suspensa.

Para armazenar dados de veículos em um bucket S3, selecione Amazon S3. S3 é um serviço de armazenamento de objetos que armazena dados como objetos em buckets. Para obter mais informações, consulte [Criar, configurar e trabalhar com buckets do Amazon S3](#) no Guia do usuário do Amazon Simple Storage Service.

O S3 otimiza o custo do armazenamento de dados e fornece mecanismos adicionais para usar dados de veículos, como data lakes, armazenamento centralizado de dados, pipelines de processamento de dados e análises. Você pode usar o S3 para armazenar dados para processamento e análise em lote. Por exemplo, você pode criar relatórios de eventos de frenagem brusca para o modelo de machine learning (ML). Os dados recebidos do veículo são armazenados em buffer por 10 minutos antes da entrega.

Amazon S3

Important

Você só pode transferir dados para o S3 se a AWS FleetWise IoT tiver permissões para gravar no bucket do S3. Para obter mais informações sobre como conceder acesso, consulte Como [controlar o acesso com a AWS IoT FleetWise](#).

Em Configurações de destino do S3, faça o seguinte:

1. Para o bucket do S3, escolha um bucket que AWS IoT FleetWise tenha permissões para.
2. (Opcional) Insira um prefixo personalizado que você possa usar para organizar os dados armazenados no bucket S3.
3. Selecione o formato de saída, que são os arquivos de formato salvos no bucket S3.
4. Selecione para compactar os dados armazenados no bucket S3 como um arquivo .gzip. Recomendamos a compactação de dados porque isso minimiza os custos de armazenamento.
5. As opções selecionadas nas configurações de destino do S3 alteram o objeto Exemplo do S3. URI Este é um exemplo de quais arquivos são salvos no S3.

Para armazenar dados de veículos em uma tabela Timestream, selecione Amazon Timestream. Você pode usar o Timestream para consultar dados do veículo para identificar tendências e padrões. Por exemplo, você pode usar o Timestream para criar um alarme para o nível de combustível do veículo. Os dados do veículo recebido são transferidos para o Timestream quase que em tempo real. Para obter mais informações, consulte [O que é o Amazon Timestream?](#) no Guia do desenvolvedor do Amazon Timestream.

Amazon Timestream

Important

Você só pode transferir dados para uma tabela se a AWS IoT FleetWise tiver permissões para gravar dados no Timestream. Para obter mais informações sobre como conceder acesso, consulte Como [controlar o acesso com a AWS IoT FleetWise](#).

Em Configurações da tabela Timestream, faça o seguinte:

1. Em Nome do banco de dados Timestream, selecione o nome do seu banco de dados Timestream na lista suspensa.
2. Em Nome da tabela Timestream, selecione o nome da tabela Timestream na lista suspensa.

Em Acesso ao serviço para Timestream, faça o seguinte:

- Escolha uma IAM função na lista suspensa.
- Escolha Próximo.

Etapa 3: adicionar veículos

Para escolher em quais veículos implantar sua campanha, selecione-os na lista de veículos. Filtre os veículos pesquisando os atributos e seus valores que você adicionou ao criar os veículos ou pelo nome do veículo.

Em Filtrar veículos, faça o seguinte:

1. Na caixa de pesquisa, encontre o atributo ou nome do veículo e escolha-o na lista.

Note

Cada atributo pode ser usado apenas uma vez.

2. Insira o valor do atributo ou o nome do veículo no qual você deseja implementar a campanha. Por exemplo, se o nome completo do atributo for fuelType, insira gasoline como o valor.
3. Para procurar outro atributo do veículo, repita as etapas anteriores. Você pode pesquisar até cinco atributos de veículos e um número ilimitado de nomes de veículos.
4. Os veículos que correspondem à sua pesquisa estão informados em Nome do veículo. Escolha os veículos nos quais deseja que a campanha seja implantada.

Note

Até 100 veículos são exibidos nos resultados da pesquisa. Escolha Selecionar tudo para adicionar todos os veículos à campanha.

5. Escolha Próximo.

Etapa 4: revisar e criar

Verifique as configurações da campanha e selecione Criar campanha.

Note

Após a criação de uma campanha, você ou sua equipe deverão implantá-la nos veículos.

Etapa 5: implantar uma campanha

Depois de criar uma campanha, você ou sua equipe deverão implantá-la nos veículos.

Para implantar uma campanha

1. Na página Resumo da campanha, selecione Implantar.
2. Revise e confirme que deseja iniciar a implantação e começar a coletar dados dos veículos conectados à campanha.
3. Escolha Implantar.

Para pausar a coleta de dados de veículos conectados à campanha, na página Resumo da campanha, selecione Suspende. Para retomar a coleta de dados dos veículos conectados à campanha, selecione Continuar.

Criar uma campanha (AWS CLI)

Você pode usar a [CreateCampaign](#) API operação para criar uma campanha. O exemplo a seguir usa a AWS CLI.

Ao criar uma campanha, os dados coletados dos veículos podem ser armazenados no Amazon S3 (S3) ou no Amazon Timestream. Selecione Timestream para obter um banco de dados de séries temporais rápido, escalável e sem servidor, e para armazenar dados que precisam de processamento quase que em tempo real. Selecione o S3 para um armazenamento de objetos com escalabilidade, disponibilidade de dados, segurança e desempenho líderes do setor.

⚠ Important

Você só pode transferir dados do veículo se a AWS IoT FleetWise tiver permissões para gravar dados no S3 ou no Timestream. Para obter mais informações sobre como conceder acesso, consulte Como [controlar o acesso com a AWS IoT FleetWise](#).

Criar campanha

⚠ Important

- Você deve ter um catálogo de sinalização e um veículo ou frota antes de criar uma campanha. Para mais informações, consulte [Gerencie AWS catálogos de sinais de IoT FleetWise](#), [Gerencie AWS veículos de IoT FleetWise](#) e [Gerencie frotas na IoT AWS FleetWise](#).
- Depois que uma campanha é criada, você deve usar a UpdateCampaign API operação para aprová-la. Para ter mais informações, consulte [Atualizar uma campanha AWS de IoT FleetWise](#)

Para criar uma campanha, execute o seguinte comando.

Substituir *file-name* com o nome do JSON arquivo que contém a configuração da campanha.

```
aws iotfleetwise create-campaign --cli-input-json file://file-name.json
```

- Substituir *campaign-name* com o nome da campanha que você está criando.
- Substituir *signal-catalog-arn* com o Amazon Resource Name (ARN) do catálogo de sinais.
- Substituir *target-arn* com a ARN de uma frota ou veículo que você criou.
- Substituir *bucket-arn* com o ARN do bucket S3.

```
{  
  "name": "campaign-name",  
  "targetArn": "target-arn",  
  "signalCatalogArn": "signal-catalog-arn",  
  "collectionScheme": {
```

```

    "conditionBasedCollectionScheme": {
      "conditionLanguageVersion": 1,
      "expression": "$variable.`Vehicle.DemoBrakePedalPressure` > 7000",
      "minimumTriggerIntervalMs": 1000,
      "triggerMode": "ALWAYS"
    }
  },
  "compression": "SNAPPY",
  "diagnosticsMode": "OFF",
  "postTriggerCollectionDuration": 1000,
  "priority": 0,
  "signalsToCollect": [
    {
      "maxSampleCount": 100,
      "minimumSamplingIntervalMs": 0,
      "name": "Vehicle.DemoEngineTorque"
    },
    {
      "maxSampleCount": 100,
      "minimumSamplingIntervalMs": 0,
      "name": "Vehicle.DemoBrakePedalPressure"
    }
  ],
  "spoolingMode": "TO_DISK",
  "dataDestinationConfigs": [
    {
      "s3Config": {
        "bucketArn": "bucket-arn",
        "dataFormat": "PARQUET",
        "prefix": "campaign-name",
        "storageCompressionFormat": "GZIP"
      }
    }
  ]
}

```

- Substituir *campaign-name* com o nome da campanha que você está criando.
- Substituir *signal-catalog-arn* com o Amazon Resource Name (ARN) do catálogo de sinais.
- Substituir *target-arn* com a ARN de uma frota ou veículo que você criou.
- Substituir *role-arn* com a função ARN de execução de tarefas que concede FleetWise permissão à AWS IoT para entregar dados à tabela Timestream.

- Substituir *table-arn* com a ARN tabela Timestream.

```
{
  "name": "campaign-name",
  "targetArn": "target-arn",
  "signalCatalogArn": "signal-catalog-arn",
  "collectionScheme": {
    "conditionBasedCollectionScheme": {
      "conditionLanguageVersion": 1,
      "expression": "$variable.`Vehicle.DemoBrakePedalPressure` > 7000",
      "minimumTriggerIntervalMs": 1000,
      "triggerMode": "ALWAYS"
    }
  },
  "compression": "SNAPPY",
  "diagnosticsMode": "OFF",
  "postTriggerCollectionDuration": 1000,
  "priority": 0,
  "signalsToCollect": [
    {
      "maxSampleCount": 100,
      "minimumSamplingIntervalMs": 0,
      "name": "Vehicle.DemoEngineTorque"
    },
    {
      "maxSampleCount": 100,
      "minimumSamplingIntervalMs": 0,
      "name": "Vehicle.DemoBrakePedalPressure"
    }
  ],
  "spoolingMode": "TO_DISK",
  "dataDestinationConfigs": [
    {
      "timestreamConfig": {
        "executionRoleArn": "role-arn",
        "timestreamTableArn": "table-arn"
      }
    }
  ]
}
```

Expressões lógicas para campanhas AWS de IoT FleetWise

AWS A IoT FleetWise usa uma expressão lógica para reconhecer quais dados coletar como parte de uma campanha. Para ter mais informações sobre expressões, consulte [Expressões](#) no Guia do desenvolvedor do AWS IoT Events .

A variável de expressão deve ser construída de acordo com as regras do tipo de dados que estão sendo coletados. Para dados do sistema de telemetria, a variável de expressão deve ser o nome totalmente qualificado do sinal. Para dados de sistemas de visão, a expressão combina o nome totalmente qualificado do sinal com o caminho que vai do tipo de dados do sinal até uma de suas propriedades.

Por exemplo, se o catálogo de sinais contém os seguintes nós:

```
{
  myVehicle.ADAS.Camera:
    type: sensor
    datatype: Vehicle.ADAS.CameraStruct
    description: "A camera sensor"

  myVehicle.ADAS.CameraStruct:
    type: struct
    description: "An obstacle detection camera output struct"
}
```

Se os nós seguirem a definição ROS 2:

```
{
  Vehicle.ADAS.CameraStruct.msg:
    boolean obstaclesExists
    uint8[] image
    Obstacle[30] obstacles
}
{
  Vehicle.ADAS.Obstacle.msg:
    float32: probability
    uint8 o_type
    float32: distance
}
```

Estas são todas as variáveis possíveis de expressão de eventos:

```
{
  ...
  $variable.`myVehicle.ADAS.Camera.obstaclesExists`
  $variable.`myVehicle.ADAS.Camera.Obstacle[0].probability`
  $variable.`myVehicle.ADAS.Camera.Obstacle[1].probability`
  ...
  $variable.`myVehicle.ADAS.Camera.Obstacle[29].probability`
  $variable.`myVehicle.ADAS.Camera.Obstacle[0].o_type`
  $variable.`myVehicle.ADAS.Camera.Obstacle[1].o_type`
  ...
  $variable.`myVehicle.ADAS.Camera.Obstacle[29].o_type`
  $variable.`myVehicle.ADAS.Camera.Obstacle[0].distance`
  $variable.`myVehicle.ADAS.Camera.Obstacle[1].distance`
  ...
  $variable.`myVehicle.ADAS.Camera.Obstacle[29].distance`
}
```

Atualizar uma campanha AWS de IoT FleetWise

Você pode usar a [UpdateCampaign](#) API operação para atualizar uma campanha existente. O comando a seguir usa AWS CLI.

- Substituir *campaign-name* com o nome da campanha que você está atualizando.
- Substituir *action* com uma das seguintes opções:
 - APPROVE— Aprova a campanha para permitir que a AWS FleetWise IoT a implemente em um veículo ou frota.
 - SUSPEND — Suspende a campanha. A campanha é excluída dos veículos, e todos os veículos na campanha suspensa deixarão de enviar dados.
 - RESUME — Reativa a campanha SUSPEND. A campanha será reimplantada em todos os veículos, e os veículos retomarão o envio de dados.
 - UPDATE — Atualiza a campanha definindo atributos e associando-os a um sinal.

```
aws iotfleetwise update-campaign \  
    --name campaign-name \  
    --action action
```

Excluir uma campanha AWS de IoT FleetWise

Você pode usar o FleetWise console de AWS IoT ou API excluir campanhas.

Excluir uma campanha (console)

Para excluir uma campanha, use o console de AWS IoT. FleetWise

Excluir uma campanha

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Campanhas.
3. Na página Campanhas, selecione a campanha alvo.
4. Escolha Excluir.
5. Em Excluir **campaign-name?**, insira o nome da campanha a ser excluída e selecione Confirmar.

Excluir uma campanha (AWS CLI)

Você pode usar a [DeleteCampaign](#) API operação para excluir uma campanha. O exemplo a seguir usa AWS CLI.

Para excluir uma campanha, execute o seguinte comando.

Substituir *campaign-name* com o nome do veículo que você está excluindo.

```
aws iotfleetwise delete-campaign --name campaign-name
```

Obtenha informações AWS da campanha de IoT FleetWise

Você pode usar a [ListCampaigns](#) API operação para verificar se uma campanha foi excluída. O exemplo a seguir usa a AWS CLI.

Execute os seguintes comandos para recuperar uma lista paginada de resumos de todas as campanhas.


```
aws iotfleetwise list-campaigns
```

Você pode usar a [GetCampaign](#) API operação para recuperar as informações do veículo. O exemplo a seguir usa a AWS CLI.

Para recuperar os metadados de uma campanha, execute o comando a seguir.

Substituir *campaign-name* com o nome da campanha que você deseja recuperar.

```
aws iotfleetwise get-campaign --name campaign-name
```

 Note

Essa operação é [eventualmente consistente](#). Em outras palavras, as alterações na campanha podem não ser refletidas imediatamente.

Visualize dados de veículos AWS de FleetWise IoT

O FleetWise software Edge Agent para AWS IoT transfere dados de veículos selecionados para o Amazon Timestream ou o Amazon Simple Storage Service (Amazon S3). Depois que seus dados chegarem ao destino dos dados, você poderá usar outros AWS serviços para visualizá-los e compartilhá-los.

Processar dados do veículo no Timestream

O Timestream é um banco de dados de séries temporais totalmente gerenciado que pode armazenar e analisar trilhões de pontos de dados de séries temporais por dia. Seus dados são armazenados em uma tabela Timestream gerenciada pelo cliente. Você pode usar o Timestream para consultar dados de veículos para obter insights sobre os veículos. Para obter mais informações, consulte [O que é o Amazon Timestream?](#)

O esquema padrão dos dados que são transferidos para o Timestream contém os seguintes campos:

Nome do campo	Tipo de dados	Descrição
eventId	varchar	O ID do evento de coleta de dados.
vehicleName	varchar	O ID do veículo do qual os dados foram coletados.
name	varchar	O nome da campanha que o software Edge Agent usa para coletar dados.
time	timestamp	A marcação de data/hora do ponto de dados.
measure_name	varchar	O nome do sinal.

Nome do campo	Tipo de dados	Descrição
measure_value::bigint	bigint	Valores de sinal do tipo Inteiro.
measure_value::double	double	Valores de sinal do tipo Duplo.
measure_value::boolean	boolean	Valores de sinal do tipo Booleano.

Visualize os dados do veículo armazenados no Timestream

Depois que os dados do veículo forem transferidos para o Timestream, você poderá usar os seguintes serviços da AWS para visualizar, monitorar, analisar e compartilhar seus dados.

- Visualize e monitore dados em painéis usando o [Grafana ou Amazon Managed Grafana](#). Você pode visualizar dados de várias AWS fontes (como Amazon CloudWatch e Timestream) e outras fontes de dados com um único painel da Grafana.
- [Analisar e visualizar dados em painéis usando a Amazon QuickSight](#)

Processar dados do veículo no Amazon S3

Amazon S3 é um serviço de armazenamento de objetos que armazena e protege qualquer quantidade de dados. Você pode usar o S3 para diversos casos de uso, como lagos de dados, backup e restauração, arquivamento, aplicativos corporativos, AWS IoT dispositivos e análise de big data. Seus dados são armazenados no S3 como objetos em buckets. Para obter mais informações, consulte [O que é a Amazon S3?](#)

O esquema padrão de dados transferidos para o Amazon S3 contém os campos a seguir.

Nome do campo	Tipo de dados	Descrição
eventId	varchar	O ID do evento de coleta de dados.

Nome do campo	Tipo de dados	Descrição
vehicleName	varchar	O ID do veículo do qual os dados foram coletados.
name	varchar	O nome da campanha que o software Edge Agent usa para coletar dados.
time	timestamp	A marcação de data/hora do ponto de dados.
measure_name	varchar	O nome do sinal.
measure_value_BIGINT	bigint	Valores de sinal do tipo Inteiro.
measure_value_DOUBLE	double	Valores de sinal do tipo Duplo.
measure_value_BOOLEAN	boolean	Valores de sinal do tipo Booleano.
measure_value_STRUCT	struct	Valores de sinal do tipo Struct.

Formato de objeto Amazon S3

AWS IoT FleetWise transfere os dados do veículo para o S3, onde são salvos como um objeto. Você pode usar o objeto URI que identifica os dados de forma exclusiva para encontrar dados da campanha. O URI formato do objeto S3 depende se os dados coletados são dados não estruturados ou processados.

Dados não estruturados

Os dados não estruturados são armazenados no S3 de uma forma não predefinida. Eles podem estar em vários formatos, como imagens ou vídeos.

As mensagens do veículo passadas para a AWS IoT FleetWise com dados de sinal dos arquivos Amazon Ion são decodificadas e transferidas para o S3 como objetos. Os objetos do S3 representam cada sinal e são codificados em binário.

O objeto S3 de dados não estruturados URI usa o seguinte formato:

```
s3://bucket-name/prefix/unstructured-data/random-ID-yyyy-MM-dd-HH-mm-ss-SSS-vehicleName-signalName-fieldName
```

Dados processados

Os dados processados são armazenados no S3 e passam por etapas de processamento que validam, enriquecem e transformam as mensagens. Listas de objetos e velocidade são exemplos de dados processados.

Os dados transferidos para o S3 são armazenados como objetos que representam registros que foram armazenados em buffer por um período de cerca de 10 minutos. Por padrão, a AWS IoT FleetWise adiciona um prefixo de UTC hora no formato `year=YYYY/month=MM/date=DD/hour=HH` antes de gravar objetos no S3. Esse prefixo cria uma hierarquia lógica no bucket, no qual cada barra (/) cria um nível na hierarquia. Os dados processados também contêm o objeto S3 URI para dados não estruturados.

O objeto S3 de dados processados URI usa o seguinte formato:

```
s3://bucket-name/prefix/processed-data/year=YYYY/month=MM/day=DD/hour=HH/part-0000-random-ID.gz.parquet
```

Dados brutos

Os dados brutos, também conhecidos como dados primários, são dados coletados dos arquivos do Amazon Ion. É possível usar dados brutos para solucionar qualquer problema ou encontrar a causa raiz dos erros.

O objeto S3 de dados brutos URI usa o seguinte formato:

```
s3://bucket-name/prefix/raw-data/vehicle-name/eventID-timestamp.10n
```

Analise os dados do veículo armazenados no Amazon S3

Depois que os dados do veículo forem transferidos para o S3, você poderá usar os serviços da AWS a seguir para monitorar, analisar e compartilhar dados.

Extraia e analise dados usando a Amazon SageMaker para fluxos de trabalho downstream de rotulagem e aprendizado de máquina (ML).

Para obter mais informações, consulte os seguintes tópicos no Amazon SageMaker Developer Guide:

- [Processar dados](#)
- [Treinar modelos de machine learning](#)
- [Rotular imagens](#)

Catalogue seus dados usando Crawler do AWS Glue e analise-os no Amazon Athena. Por padrão, os objetos gravados no S3 têm partições de tempo no estilo Apache Hive, com caminhos de dados que contêm pares de valores-chave conectados por sinais de igual.

Para obter mais informações, consulte os seguintes tópicos no Guia do usuário do Amazon Athena:

- [Particionamento de dados no Athena](#)
- [Usando AWS Glue para se conectar a fontes de dados no Amazon S3](#)
- [Melhores práticas ao usar o Athena com AWS Glue](#)

Visualize dados usando a Amazon QuickSight lendo diretamente sua tabela do Athena ou bucket do S3.

Tip

Se você estiver lendo diretamente do S3, confirme se os dados do seu veículo estão no JSON formato porque a Amazon QuickSight não suporta o formato Apache Parquet.

Para obter mais informações, consulte os seguintes tópicos no Guia do QuickSight usuário da Amazon:

- [Fonte de dados compatíveis](#)
- [Criação de fonte de dados](#)

Use AWS CLI e AWS SDKs com a AWS IoT FleetWise

Esta seção fornece informações sobre como fazer solicitações de AWS IoT. FleetWise API Para obter mais informações sobre FleetWise [operações e tipos de dados](#) de AWS IoT, consulte a Referência de IoT AWS . FleetWise API

Para usar a AWS IoT FleetWise com uma variedade de linguagens de programação, use o [AWS SDKs](#), que contém as seguintes funcionalidades automáticas:

- Assinar criptograficamente suas solicitações de serviço
- Recuperar solicitações
- Lidar com respostas de erro

Para acesso à linha de comando, use a AWS IoT FleetWise com o [AWS CLI](#) Você pode controlar a AWS IoT FleetWise e seus outros serviços a partir da linha de comando e automatizá-los por meio de scripts.

Solução de problemas de AWS IoT FleetWise

Use as informações e soluções de solução de problemas nesta seção para ajudar a resolver problemas com a AWS IoT FleetWise.

As informações a seguir podem ajudá-lo a solucionar problemas comuns com a AWS IoT FleetWise.

Tópicos

- [AWS Problemas manifestos do FleetWise decodificador de IoT](#)
- [Edge Agent para problemas AWS de software de IoT FleetWise](#)

AWS Problemas manifestos do FleetWise decodificador de IoT

Solucione problemas com manifesto do decodificador.

Diagnosticando chamadas de manifesto do decodificador API

Erro	Diretrizes de solução de problemas
<code>UpdateOperationFailure.ConflictingDecoderUpdate</code>	O mesmo manifesto do decodificador tem várias solicitações de atualização. Espere e tente novamente.
<code>UpdateOperationFailure.InternalFailure</code>	<code>InternalFailure</code> é lançado como uma exceção encapsulada. O problema em si depende da exceção encapsulada.
<code>UpdateOperationFailure.ActiveDecoderUpdate</code>	O manifesto do decodificador está em um estado <code>Active</code> e não pode ser atualizado. Altere o estado do manifesto do decodificador para <code>DRAFT</code> e tente novamente.
<code>UpdateOperationFailure.ConflictingModelUpdate</code>	AWS A IoT FleetWise está tentando se validar em relação a um modelo de veículo (manifesto do modelo) que está sendo modificado por outra pessoa. Espere e tente novamente.

Erro	Diretrizes de solução de problemas
<pre>UpdateOperationFailure.Mode ManifestValidationResponse : FailureReason.MODEL_DATA_ENTRIES_NOT_FOUND</pre>	<p>O modelo do veículo não tem nenhum sinal associado a ele. Adicione sinais ao modelo do veículo e verifique se eles podem ser encontrados no catálogo de sinais associado.</p>
<pre>UpdateOperationFailure.Mode ManifestValidationResponse : FailureReason.MODEL_NOT_ACTIVE</pre>	<p>Atualize o modelo do veículo para que ele fique no estado ACTIVE e tente novamente.</p>
<pre>UpdateOperationFailure.Mode ManifestValidationResponse : FailureReason.MODEL_NOT_FOUND</pre>	<p>AWS A IoT não FleetWise consegue encontrar o modelo do veículo associado ao manifesto do decodificador. Verifique o Amazon Resource Name (ARN) do modelo do veículo e tente novamente.</p>
<pre>UpdateOperationFailure.Mode ManifestValidationResponse (FailureReason.MODEL_DATA_ENTRIES_READ_FAILURE)</pre>	<p>A validação do modelo do veículo falhou porque os nomes dos sinais do modelo do veículo não foram encontrados no catálogo de sinais. Verifique se os sinais no modelo do veículo estão todos incluídos no catálogo de sinais associado.</p>
<pre>UpdateOperationFailure.ValidationFailure</pre>	<p>Sinais ou interfaces de rede que não são válidos foram encontrados na solicitação para atualizar o manifesto do decodificador. Verifique se todos os sinais e interfaces de rede retornados pela exceção existem, se todos os sinais usados estão associados a uma interface disponível e se você não removerá uma interface que tenha sinais associados a ela.</p>

Erro	Diretrizes de solução de problemas
<code>UpdateOperationFailure.KmsKeyAccessDenied</code>	Há um problema de permissão na tecla AWS Key Management Service (AWS KMS) usada para a operação. Verifique se você está usando um perfil que tenha acesso à chave e tente novamente.
<code>UpdateOperationFailure.DecoderDoesNotExist</code>	O manifesto do decodificador não existe. Verifique o nome manifesto do decodificador e tente novamente.

As mensagens de erro de dados de sistemas de visão com o motivo `SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG` incluirão uma dica na resposta que fornece informações sobre por que a solicitação falhou. Você pode usar a dica para determinar quais diretrizes de solução de problemas seguir.

Note

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Diagnosticar a validação de dados de sistemas de visão do manifesto do decodificador

Erro	Diretrizes de solução de problemas
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.NO_SIGNAL_IN_CATALOG_FOR_DECODER_SIGNAL)</code>	AWS A IoT FleetWise não encontrou a estrutura do sinal raiz usada no decodificador de sinal usando o catálogo de sinais. Verifique se o sinal raiz da estrutura está definido corretamente no catálogo de sinais.
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_TYPE_INCOMPATIBLE_WITH_MESSAGE_SIGNAL_TYPE)</code>	Uma mensagem primitiva no catálogo de sinais não foi definida com o mesmo tipo de dados na solicitação de atualização do manifesto do decodificador. Verifique se as mensagens primitivas definidas na solicitação

Erro	Diretrizes de solução de problemas
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.STRUCT_SIZE_MISMATCH)</code>	<p>correspondem à definição do catálogo de sinais correspondente.</p> <p>O número de propriedades definidas em uma struct no catálogo de sinais não corresponde ao número de propriedades que você está tentando decodificar no manifesto do decodificador. Verifique se você tem o número correto de sinais para decodificar comparando-o com os sinais definidos no catálogo de sinais.</p>
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</code>	<p>AWS A IoT FleetWise encontrou um sinal definido como a STRUCT no catálogo de sinais sem um <code>structuredMessageDefinition</code> definido na solicitação do manifesto do decodificador. Certifique-se de que cada estrutura esteja definida como a <code>structuredMessageDefinition</code> na solicitação de atualização do manifesto do decodificador.</p>
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</code>	<p>O sinal raiz da estrutura usada no manifesto do decodificador não está definido adequadamente como uma estrutura no catálogo de sinais. A estrutura do sinal raiz usada no manifesto do decodificador deve ter seu <code>structFullyQualified</code> nome de campo definido. Ele também precisa de um STRUCT nó com isso <code>fullyQualifiedName</code>.</p>
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</code>	<p>Uma das mensagens de folha usadas na solicitação do manifesto do decodificador não está definida como uma mensagem primitiva. Verifique se todos os objetos de folha na solicitação estão definidos como mensagens primitivas.</p>

Erro	Diretrizes de solução de problemas
<pre>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</pre>	<p>Um objeto de matriz no catálogo de sinais não foi definido como uma <code>structuredMessageList</code> na solicitação de atualização do manifesto do decodificador. Verifique se todas as propriedades da matriz estão definidas como <code>structuredMessageList</code> na solicitação de atualização do manifesto do decodificador.</p>

Edge Agent para problemas AWS de software de IoT FleetWise

Solucione problemas do software Edge Agent.

Problemas

- [Problema: o software Edge Agent não inicia.](#)
- [Problema: \[ERROR\] \[IoT FleetWise Engine: :connect\]: \[Falha ao iniciar a biblioteca de persistência\]](#)
- [Problema: O software Edge Agent não coleta diagnósticos integrados \(OBD\) II PIDs e códigos de problemas de diagnóstico \(DTCs\).](#)
- [Problema: o FleetWise software Edge Agent para AWS IoT não coleta dados da rede ou não consegue aplicar regras de inspeção de dados.](#)
- [Problema: \[ERROR\] \[AwsIotConnectivityModule: :connect\]: \[Falha na conexão com erro\] ou \[WARN\] \[AwsIotChannel: :send\]: \[Nenhuma MQTT conexão ativa.\]](#)

Problema: o software Edge Agent não inicia.

Pode ser que você veja os seguintes erros quando o software Edge Agent não inicia.

- ```
Error from reader: * Line 1, Column 1
Syntax error: value, object or array expected.
```

Solução: verifique se o arquivo de configuração do FleetWise software Edge Agent para AWS IoT está usando um formato válido JSON. Por exemplo, certifique-se de que as vírgulas estão sendo

usadas de forma correta. Para obter mais informações sobre o arquivo de configuração, faça o seguinte para baixar o Guia do desenvolvedor do FleetWise software Edge Agent para AWS IoT.

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. Na página inicial do serviço, na FleetWise seção Introdução à AWS IoT, escolha Explore o Edge Agent.

```
[ERROR] [SocketCANBusChannel::connect]: [SocketCan with name xxx is not accessible]
[ERROR] [IoTFleetWiseEngine::connect]: [Failed to Bind Consumers to Producers]
```

Solução: esse erro pode ser exibido quando o software Edge Agent não consegue estabelecer comunicação de soquete com as interfaces de rede definidas no arquivo de configuração.

Para verificar se todas as interfaces de rede definidas na configuração estão disponíveis, execute o seguinte comando:

```
ip link show
```

Para colocar uma interface de rede online, execute o seguinte comando: Substituir *network-interface-id* com o ID da interface de rede.

```
sudo ip link set network-interface-id up
```

```
[ERROR] [AwsIotConnectivityModule::connect]: [Connection failed with error]
[WARN] [AwsIotChannel::send]: [No alive MQTT Connection.]
or
[WARN] [AwsIotChannel::send]: [aws-c-common: AWS_ERROR_FILE_INVALID_PATH]
```

Solução: Você pode ver esse erro quando o software Edge Agent não consegue estabelecer uma MQTT conexão com AWS IoT Core o. Verifique se os itens a seguir estão configurados corretamente e reinicie o software Edge Agent.

- `mqtConnection::endpointUrl`— AWS endpoint do dispositivo IoT da conta.
- `mqtConnection::clientId`— O ID do veículo no qual o software Edge Agent está em execução.
- `mqtConnection::certificateFilename`— O caminho para o arquivo de certificado do veículo.

- `mqttConnection::privateKeyFilename`— O caminho para o arquivo da chave privada do veículo.
- Você costumava AWS IoT Core provisionar o veículo. Para obter mais informações, consulte [Provisionar AWS veículos de IoT FleetWise](#).

Para obter mais informações, consulte [Perguntas frequentes sobre o AWS IoT Device SDK for C++](#).

## Problema: [ERROR] [IoT FleetWise Engine: :connect]: [Falha ao iniciar a biblioteca de persistência]

Solução: esse erro pode ser exibido quando o software Edge Agent não consegue localizar o armazenamento de persistência. Verifique se o seguinte está configurado corretamente e reinicie o software Edge Agent.

`persistence:persistencePath` — Um caminho local usado para persistir esquemas de coleta, manifestos de decodificadores e instantâneos de dados.

## Problema: O software Edge Agent não coleta diagnósticos integrados (OBD) II PIDs e códigos de problemas de diagnóstico (DTCs).

Solução: esse erro pode ser exibido quando `obdInterface:pidRequestIntervalSeconds` ou `obdInterface:dtcRequestIntervalSeconds` estão configurados como 0.

Se o software Edge Agent estiver sendo executado em um veículo com transmissão automática, verifique se `obdInterface:hasTransmissionEcu` está configurado para `true`.

Se o seu veículo suportar a arbitragem estendida da Rede de Área do Controlador (CANbus)IDs, verifique se `obdInterface:useExtendedIds` está configurado para `true`.

## Problema: o FleetWise software Edge Agent para AWS IoT não coleta dados da rede ou não consegue aplicar regras de inspeção de dados.

Solução: esse erro pode ser exibido quando as cotas padrão são violadas.

| Recurso                                                   | Quota                                          | Ajustável | Observação                                                                                                                                                                           |
|-----------------------------------------------------------|------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Valor do ID do sinal                                      | O ID do sinal deve ser menor ou igual a 50.000 | Sim       | O software Edge Agent não coletará dados de sinais que tenham um ID superior a 50.000. Recomendamos verificar quantos sinais o catálogo de sinais contém antes de alterar essa cota. |
| Número de esquemas ativos de recolha de dados por veículo | 256                                            | Sim       | Recomendamos que você verifique quantas campanhas criou na nuvem e quantos esquemas cada campanha contém antes de alterar essa cota.                                                 |
| Tamanho do buffer do histórico do sinal                   | 20 MB                                          | Sim       | Se a cota for violada, o software Edge Agent para de coletar novos dados.                                                                                                            |

**Problema:** [ERROR] [AwsIotConnectivityModule: :connect]: [Falha na conexão com erro] ou [WARN] [AwsIotChannel: :send]: [Nenhuma MQTT conexão ativa.]

**Solução:** esse erro pode ser exibido quando o software Edge Agent não está conectado à nuvem. Por padrão, o software Edge Agent envia uma solicitação de ping a AWS IoT Core cada minuto e espera por três minutos. Se não houver resposta, o software Edge Agent restabelecerá automaticamente a conexão com a nuvem.

# Segurança na AWS IoT FleetWise

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como a segurança da nuvem e a segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de [AWS](#) de . Para saber mais sobre os programas de conformidade que se aplicam à AWS IoT FleetWise, consulte [AWS Serviços no escopo do programa de conformidade AWS Serviços no escopo do programa](#) .
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade dos dados, os requisitos da empresa e as leis e os regulamentos aplicáveis

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar a AWS IoT FleetWise. Ele mostra como configurar a AWS IoT para atender FleetWise aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos de AWS IoT FleetWise .

## Conteúdo

- [Proteção de dados na AWS IoT FleetWise](#)
- [Controlando o acesso com AWS IoT FleetWise](#)
- [Identity and Access Management para AWS IoT FleetWise](#)
- [Validação de conformidade para AWS IoT FleetWise](#)
- [Resiliência na IoT AWS FleetWise](#)
- [Segurança da infraestrutura na AWS IoT FleetWise](#)
- [Análise de configuração e vulnerabilidade na AWS IoT FleetWise](#)
- [Melhores práticas de segurança para AWS IoT FleetWise](#)

# Proteção de dados na AWS IoT FleetWise

O [modelo de responsabilidade AWS compartilhada](#) de se aplica à proteção de dados na AWS IoT FleetWise. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre privacidade de dados, consulte [Privacidade de dados FAQ](#). Para obter informações sobre proteção de dados na Europa, consulte o [Modelo de Responsabilidade AWS Compartilhada e GDPR](#) a postagem no blog AWS de segurança.

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use a autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos. Exigimos TLS 1,2 e recomendamos TLS 1,3.
- Configure API e registre as atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como [trabalhar com CloudTrail trilhas](#) no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de FIPS 140-3 módulos criptográficos validados ao acessar AWS por meio de uma interface de linha de comando ou uma API, use um endpoint. FIPS Para obter mais informações sobre os FIPS endpoints disponíveis, consulte [Federal Information Processing Standard \(FIPS\) 140-3](#).

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de e-mail dos seus clientes, em marcações ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com AWS IoT FleetWise ou outra coisa Serviços da AWS usando o console,, API AWS CLI, ou. AWS SDKs Quaisquer dados inseridos em tags ou



campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é altamente recomendável que você não inclua informações de credenciais no URL para validar sua solicitação para esse servidor.

AWS A IoT FleetWise deve ser usada com um Edge Agent que você desenvolve e instala no hardware compatível do veículo para transmitir dados do veículo para a AWS nuvem. A extração de dados de veículos pode estar sujeita a regulamentos de privacidade de dados em determinadas jurisdições. Antes de usar a AWS IoT FleetWise e instalar seu Edge Agent, é altamente recomendável que você avalie suas obrigações de conformidade de acordo com a lei aplicável. Isso inclui requisitos legais aplicáveis para fornecer avisos de privacidade legalmente adequados e obter os consentimentos necessários para extrair dados do veículo.

## Criptografia em repouso na AWS IoT FleetWise

Os dados coletados de um veículo são transmitidos para a nuvem por meio de uma AWS IoT Core mensagem com o protocolo de MQTT mensagem. AWS A IoT FleetWise entrega os dados ao seu banco de dados Amazon Timestream. No Timestream, seus dados são criptografados. Todos Serviços da AWS criptografam dados em repouso por padrão.

A criptografia em repouso se integra com AWS Key Management Service (AWS KMS) para gerenciar a chave de criptografia usada para criptografar seus dados. Você pode optar por usar uma chave gerenciada pelo cliente para criptografar os dados coletados pela AWS IoT FleetWise. Você pode criar, gerenciar e visualizar sua chave de criptografia por meio de AWS KMS. Para obter mais informações, consulte [O que é AWS Key Management Service?](#) no Guia do AWS Key Management Service desenvolvedor.

## Criptografia em trânsito

Todos os dados trocados com AWS IoT serviços são criptografados em trânsito usando Transport Layer Security (TLS). Para obter mais informações, consulte [Segurança de transporte](#) no Guia do desenvolvedor do AWS IoT .

Além disso, AWS IoT Core oferece suporte à [autenticação](#) e [autorização](#) para ajudar a controlar com segurança o acesso aos recursos de IoT AWS . FleetWise Os veículos podem usar certificados X.509 para serem autenticados (conectados) para usar a AWS IoT FleetWise e usar AWS IoT Core políticas para serem autorizados (ter permissões) para realizar ações específicas. Para obter mais informações, consulte [the section called “Provisionar veículos”](#).

# Criptografia de dados na AWS IoT FleetWise

A criptografia de dados se refere à proteção de dados em trânsito (à medida que viajam de e para a AWS FleetWise IoT e entre gateways e servidores) e em repouso (enquanto são armazenados em dispositivos locais ou em). Serviços da AWS Você pode proteger os dados em repouso usando criptografia do lado do cliente.

## Note

AWS O processamento de FleetWise borda da IoT expõe APIs que estão hospedadas em FleetWise gateways de AWS IoT e podem ser acessadas pela rede local. Eles APIs são expostos por meio de uma TLS conexão apoiada por um certificado de servidor de propriedade do conector IoT Edge AWS . FleetWise Para autenticação do cliente, eles APIs usam uma senha de controle de acesso. A chave privada do certificado do servidor e a senha de controle de acesso são armazenadas em disco. AWS O processamento de FleetWise borda da IoT depende da criptografia do sistema de arquivos para a segurança dessas credenciais em repouso.

Para obter mais informações sobre criptografia do lado do servidor e criptografia do lado do cliente, consulte os tópicos a seguir.

## Conteúdo

- [Criptografia em repouso na AWS IoT FleetWise](#)
- [Gerenciamento de chaves na AWS IoT FleetWise](#)

## Criptografia em repouso na AWS IoT FleetWise

AWS A IoT FleetWise armazena seus dados na AWS nuvem e em gateways.

### Dados em repouso na AWS nuvem

AWS A IoT FleetWise armazena dados em outros Serviços da AWS que criptografam dados em repouso por padrão. A criptografia em repouso se integra com [AWS Key Management Service \(AWS KMS\)](#) para gerenciar a chave de criptografia usada para criptografar os valores das propriedades do seu ativo e agregar valores na IoT. AWS FleetWise Você pode optar por usar uma chave gerenciada pelo cliente para criptografar valores de propriedades de ativos e valores agregados na AWS IoT. FleetWise Você pode criar, gerenciar e visualizar sua chave de criptografia por meio de AWS KMS.

Você pode escolher uma chave gerenciada pelo cliente Chave pertencente à AWS ou uma chave gerenciada pelo cliente para criptografar seus dados.

### Como funciona

A criptografia em repouso se AWS KMS integra ao gerenciamento da chave de criptografia usada para criptografar seus dados.

- Chave pertencente à AWS — Chave de criptografia padrão. AWS A IoT FleetWise possui essa chave. Você não pode visualizar, gerenciar ou usar essa chave em Conta da AWS. Você também não pode ver as operações na chave nos AWS CloudTrail registros. Você pode usar esta chave sem custo adicional.
- Chave gerenciada pelo cliente – A chave é armazenada na sua conta e é você que a cria, detém e gerencia. Você tem controle total sobre a KMS chave. AWS KMS Taxas adicionais se aplicam.

### Chaves pertencentes à AWS

Chaves pertencentes à AWS não estão armazenados em sua conta. Eles fazem parte de uma coleção de KMS chaves que AWS possui e gerencia para uso em várias Contas da AWS. Serviços da AWS pode ser usado Chaves pertencentes à AWS para proteger seus dados.

Você não pode visualizar, gerenciar Chaves pertencentes à AWS, usar ou auditar seu uso. No entanto, você não precisa realizar nenhuma ação nem alterar nenhum programa para proteger as chaves que criptografam seus dados.

Você não pagará nenhuma taxa se usar Chaves pertencentes à AWS, e elas não contam nas AWS KMS cotas da sua conta.

### Chaves gerenciadas pelo cliente

As chaves gerenciadas pelo cliente são KMS chaves em sua conta que você cria, possui e gerencia. Você tem controle total sobre essas KMS chaves, como as seguintes:

- Estabelecendo e mantendo suas principais políticas, IAM políticas e subsídios
- Ativação e desativação das chaves
- Alternar os materiais de criptografia das chaves
- Adicionar etiquetas
- Criar aliases que se referem as chaves

- Agendá-las para exclusão

Você também pode usar o CloudTrail Amazon CloudWatch Logs para rastrear as solicitações que a AWS IoT FleetWise envia AWS KMS em seu nome.

Se você estiver usando chaves gerenciadas pelo cliente, deverá conceder FleetWise acesso de AWS IoT à KMS chave armazenada em sua conta. A AWS IoT FleetWise usa criptografia de envelope e hierarquia de chaves para criptografar dados. A chave de criptografia do AWS KMS é usada para criptografar a chave raiz dessa hierarquia de chaves. Para obter mais informações, consulte [Criptografia envelopada](#) no Guia do desenvolvedor do AWS Key Management Service .

O exemplo de política a seguir concede FleetWise permissões de AWS IoT para criar uma chave gerenciada pelo cliente em seu nome.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "Stmt1603902045292",
 "Action": [
 "kms:GenerateDataKey*",
 "kms:Decrypt",
 "kms:DescribeKey",
 "kms:CreateGrant",
 "kms:RetireGrant",
 "kms:RevokeGrant"
],
 "Effect": "Allow",
 "Resource": "*"
 }
]
}
```

#### Important

Ao adicionar as novas seções à sua política de KMS chaves, não altere nenhuma seção existente na política. A AWS IoT não FleetWise pode realizar operações em seus dados se a criptografia estiver habilitada para AWS IoT FleetWise e se qualquer uma das afirmações a seguir for verdadeira:

- A KMS chave está desativada ou excluída.
- A política de KMS chaves não está configurada corretamente para o serviço.

## Usar dados de sistemas de visão com criptografia em repouso

### Note

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Se você tiver criptografia gerenciada pelo cliente com AWS KMS chaves habilitadas em sua FleetWise conta de AWS IoT e quiser usar dados do sistema de visão, redefina suas configurações de criptografia para serem compatíveis com tipos de dados complexos. Isso permite que a AWS IoT FleetWise estabeleça as permissões adicionais necessárias para os dados do sistema de visão.

### Note

O manifesto do decodificador poderá ficar preso em um status de validação se você não tiver redefinido as configurações de criptografia dos dados de sistemas de visão.

1. Use a [GetEncryptionConfiguration](#) API operação para verificar se a AWS KMS criptografia está ativada. Nenhuma ação adicional será necessária se o tipo de criptografia for `FLEETWISE_DEFAULT_ENCRYPTION`.
2. Se o tipo de criptografia for `KMS_BASED_ENCRYPTION`, use a [PutEncryptionConfiguration](#) API operação para redefinir o tipo de criptografia para `FLEETWISE_DEFAULT_ENCRYPTION`.

```
{
 aws iotfleetwise put-encryption-configuration --encryption-type
 FLEETWISE_DEFAULT_ENCRYPTION
}
```

3. Use a [PutEncryptionConfiguration](#) API operação para reativar o tipo de criptografia para `KMS_BASED_ENCRYPTION`.

```
{
```

```
aws iotfleetwise put-encryption-configuration \
 --encryption-type "KMS_BASED_ENCRYPTION"
 --kms-key-id kms_key_id
}
```

Para ter mais informações sobre como habilitar a criptografia, consulte [Gerenciamento de chaves na AWS IoT FleetWise](#).

## Gerenciamento de chaves na AWS IoT FleetWise


### AWS Gerenciamento de chaves na FleetWise nuvem de IoT

Por padrão, a AWS IoT FleetWise usa Chaves gerenciadas pela AWS para proteger seus dados no. Nuvem AWS Você pode atualizar suas configurações para usar uma chave gerenciada pelo cliente para criptografar dados na AWS FleetWise IoT. Você pode criar, gerenciar e visualizar sua chave de criptografia por meio de AWS Key Management Service (AWS KMS).

AWS A IoT FleetWise oferece suporte à criptografia do lado do servidor com chaves gerenciadas pelo cliente armazenadas AWS KMS para criptografar dados para os seguintes recursos.

| AWS Recurso de IoT FleetWise            | Tipo de dados | Campos criptografados em repouso com chaves gerenciadas pelo cliente |
|-----------------------------------------|---------------|----------------------------------------------------------------------|
| Catálogo de sinais                      |               | description                                                          |
|                                         | Atributo      | descrição,allowedValues,defaultValue, min, max                       |
|                                         | Actuator      | descriçãoallowedValues, min, max                                     |
|                                         | Sensor        | descriçãoallowedValues, min, max                                     |
| Modelo do veículo (manifesto do modelo) |               | description                                                          |
| Manifesto do decodificador              |               | description                                                          |
|                                         | CanInterface  | protocolName, protocolVersion                                        |

| AWS Recurso de IoT FleetWise | Tipo de dados                   | Campos criptografados em repouso com chaves gerenciadas pelo cliente                                                                   |
|------------------------------|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
|                              | ObdInterface                    | requestMessageId, dtcReques<br>tInterval Segundos, hasTransm<br>issionEcu,obdStandard, pidReques<br>tInterval Segundos, useExtendedIds |
|                              | CanSignal                       | fator, isBigEndian,isSigned,<br>comprimentomessageId, deslocame<br>nto, startBit                                                       |
|                              | ObdSignal                       | byteLength, offset, pid, escalabil<br>idade pidResponseLength,,service<br>Mode, startByte bitMaskLength<br>bitRightShift               |
| Veículo                      |                                 | attributes                                                                                                                             |
| Campanha                     |                                 | description                                                                                                                            |
|                              | conditionBasedCollectionEsquema | expressão conditionLanguageV<br>ersion, minimumTriggerInterval<br>Senhora, triggerMode                                                 |
|                              | TimeBasedCollectionScheme       | periodMs                                                                                                                               |

 Note

Outros dados e recursos são criptografados usando a criptografia padrão com chaves gerenciadas pela AWS IoT FleetWise. Essa chave é criada e armazenada na conta de AWS IoT. FleetWise

Para obter mais informações, consulte [O que é AWS Key Management Service?](#) no Guia do AWS Key Management Service desenvolvedor.

## Ativar criptografia usando KMS chaves (console)

Para usar chaves gerenciadas pelo cliente com a AWS IoT FleetWise, você deve atualizar suas configurações de IoT AWS . FleetWise

Para ativar a criptografia usando KMS chaves (console)

1. Abra o console [AWS de IoT. FleetWise](#)
2. Acesse Configurações.
3. Em Criptografia, selecione Editar para abrir a página Editar criptografia.
4. Em Tipo de chave de criptografia, escolha Escolher uma AWS KMS chave diferente. Isso permite a criptografia com chaves gerenciadas pelo cliente armazenadas na AWS KMS.

### Note

Você só pode usar a criptografia de chave gerenciada pelo cliente para recursos de AWS IoT FleetWise . Isso inclui o catálogo de sinais, modelo do veículo (manifesto do modelo), manifesto do decodificador, veículo, frota e campanha.

5. Escolha sua KMS chave com uma das seguintes opções:
  - Para usar uma KMS chave existente — Escolha seu alias de KMS chave na lista.
  - Para criar uma nova KMS chave, escolha Criar uma AWS KMS chave.

### Note

Isso abre o AWS KMS console. Para obter mais informações sobre como criar uma KMS chave, consulte [Criação de chaves](#) no Guia do AWS Key Management Service desenvolvedor.

6. Selecione Salvar para atualizar as configurações.

## Ativar criptografia usando KMS chaves (AWS CLI)

Você pode usar a [PutEncryptionConfiguration](#) API operação para ativar a criptografia para sua conta de AWS IoT FleetWise . O exemplo a seguir usa AWS CLI.

Para ativar a criptografia, execute o seguinte comando.



- Substituir *KMS key id* com o ID da KMS chave.

```
aws iotfleetwise put-encryption-configuration --kms-key-id KMS key id --encryption-type
KMS_BASED_ENCRYPTION
```

### Example Retorno

```
{
 "kmsKeyId": "customer_kms_key_id",
 "encryptionStatus": "PENDING",
 "encryptionType": "KMS_BASED_ENCRYPTION"
}
```

### KMS política chave

Depois de criar uma KMS chave, você deve, no mínimo, adicionar a seguinte declaração à sua política de KMS chaves para que ela funcione com a AWS IoT FleetWise.

```
{
 "Sid": "Allow FleetWise to encrypt and decrypt data when customer managed KMS key
based encryption is enabled",
 "Effect": "Allow",
 "Principal": {
 "Service": "iotfleetwise.amazonaws.com"
 },
 "Action": [
 "kms:GenerateDataKey*",
 "kms:Decrypt",
 "kms:DescribeKey",
 "kms:CreateGrant",
 "kms:RetireGrant",
 "kms:RevokeGrant"
],
 "Resource": "*"
}
```

Para obter mais informações sobre como editar uma política de KMS chaves para uso com a AWS IoT FleetWise, consulte [Alterar uma política de chaves](#) no Guia do AWS Key Management Service desenvolvedor.

**⚠ Important**

Ao adicionar as novas seções à sua política de KMS chaves, não altere nenhuma seção existente na política. AWS IoT FleetWise não pode realizar operações em seus dados se a criptografia estiver habilitada para AWS IoT FleetWise e se qualquer uma das afirmações a seguir for verdadeira:

- A KMS chave está desativada ou excluída.
- A política de KMS chaves não está configurada corretamente para o serviço.

## Controlando o acesso com AWS IoT FleetWise

As seções a seguir abordam como controlar o acesso de e para seus AWS IoT FleetWise recursos. As informações que eles abrangem incluem como conceder acesso ao seu aplicativo para que a AWS IoT FleetWise possa transferir dados do veículo durante as campanhas. Eles também descrevem como você pode conceder AWS IoT FleetWise acesso ao seu bucket Amazon S3 (S3) ou ao banco de dados e tabela do Amazon Timestream para armazenar dados.

A tecnologia para gerenciar todas essas formas de acesso é AWS Identity and Access Management (IAM). Para obter mais informações sobre IAM, consulte [O que é IAM?](#).

### Conteúdo

- [Conceder AWS IoT FleetWise acesso a um destino do Amazon S3](#)
- [Conceder AWS IoT FleetWise acesso a um destino do Amazon Timestream](#)

## Conceder AWS IoT FleetWise acesso a um destino do Amazon S3

Quando você usa um destino do Amazon S3, AWS IoT FleetWise entrega os dados do veículo para o seu bucket do S3 e, opcionalmente, pode usar uma AWS KMS chave que você possui para criptografia de dados. Se o registro de erros estiver ativado, AWS IoT FleetWise também enviará erros de entrega de dados para seu grupo de CloudWatch registros e fluxos. Você precisa ter uma IAM função ao criar um stream de entrega.

AWS IoT FleetWise usa uma política de bucket com o principal de serviço para o destino S3. Para obter mais informações sobre como adicionar políticas de bucket, consulte [Adicionar uma política de bucket usando o console do Amazon S3](#) no Guia do usuário do Amazon Simple Storage Service.

Use a política de acesso a seguir para permitir o acesso AWS IoT FleetWise ao seu bucket do S3. Se você não tiver o bucket do S3, adicione `s3:PutObjectAcl` à lista de ações do Amazon S3. Isso concede ao proprietário do bucket acesso total aos objetos entregues por AWS IoT FleetWise. Para obter mais informações sobre como proteger o acesso a objetos em seus buckets, consulte [Exemplos de políticas de bucket](#) no Guia do usuário do Amazon Simple Storage Service.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Service": [
 "iotfleetwise.amazonaws.com"
]
 },
 "Action": [
 "s3:ListBucket"
],
 "Resource": "arn:aws:s3:::bucket-name"
 },
 {
 "Effect": "Allow",
 "Principal": {
 "Service": [
 "iotfleetwise.amazonaws.com"
]
 },
 "Action": [
 "s3:GetObject",
 "s3:PutObject"
],
 "Resource": "arn:aws:s3:::bucket-name/*",
 "Condition": {
 "StringEquals": {
 "aws:SourceArn": "campaign-arn",
 "aws:SourceAccount": "account-id"
 }
 }
 }
]
}
```

A política de bucket a seguir é para todas as campanhas em uma conta em uma AWS região.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Service": [
 "iotfleetwise.amazonaws.com"
]
 },
 "Action": [
 "s3:ListBucket"
],
 "Resource": "arn:aws:s3:::bucket-name"
 },
 {
 "Effect": "Allow",
 "Principal": {
 "Service": [
 "iotfleetwise.amazonaws.com"
]
 },
 "Action": [
 "s3:GetObject",
 "s3:PutObject"
],
 "Resource": "arn:aws:s3:::bucket-name/*",
 "Condition": {
 "StringLike": {
 "aws:SourceArn": "arn:aws:iotfleetwise:region:account-id:campaign/*",
 "aws:SourceAccount": "account-id"
 }
 }
 }
]
}
```

Se você tiver uma KMS chave anexada ao seu bucket do S3, a chave precisará da seguinte política. Para obter informações sobre gerenciamento de chaves, consulte [Proteção de dados usando criptografia do lado do servidor com AWS Key Management Service chaves \(SSE-KMS\) no Guia do usuário do Amazon Simple Storage Service](#).

```
{
 "Version": "2012-10-17",
 "Effect": "Allow",
 "Principal": {
 "Service": "iotfleetwise.amazonaws.com"
 },
 "Action": [
 "kms:GenerateDataKey",
 "kms:Decrypt"
],
 "Resource": "key-arn"
}
```

### Important

Quando você cria um bucket, o S3 cria uma lista de controle de acesso padrão (ACL) que concede ao proprietário do recurso controle total sobre o recurso. Se a AWS IoT não FleetWise puder fornecer dados para o S3, certifique-se de desativá-los ACL no bucket do S3. Para obter mais informações, consulte [Desativação ACLs de todos os novos buckets e imposição da propriedade de objetos no Guia do usuário](#) do Amazon Simple Storage Service.

## Conceder AWS IoT FleetWise acesso a um destino do Amazon Timestream

Quando você usa um destino Timestream, AWS IoT FleetWise entrega os dados do veículo para uma tabela Timestream. Você deve anexar as políticas à IAM função para permitir o envio de dados AWS IoT FleetWise para o Timestream.

Se você usa o console para [criar uma campanha](#), a AWS IoT anexa FleetWise automaticamente a política necessária à função.

Antes de começar, verifique o seguinte:

### Important

- Você deve usar a mesma AWS região ao criar recursos de Timestream para IoT AWS . FleetWise Se você mudar de AWS região, poderá ter problemas para acessar os recursos do Timestream.

- AWS A IoT FleetWise está disponível no Leste dos EUA (Norte da Virgínia) e na Europa (Frankfurt).
  - Para ver a lista de regiões compatíveis, consulte [Endpoints e cotas do Timestream](#) na Referência geral da AWS.
- 
- É necessário ter um banco de dados do Timestream. Para ver um tutorial, consulte [Criar um banco de dados](#) no Guia do desenvolvedor do Amazon Timestream.
  - Você deve ter uma tabela criada no banco de dados do Timestream especificado. Para ver um tutorial, consulte [Criar uma tabela](#) no Guia do desenvolvedor do Amazon Timestream.

Você pode usar o AWS CLI para criar uma IAM função com uma política de confiança para Timestream. Para criar uma IAM função, execute o comando a seguir.

Para criar uma IAM função com uma política de confiança

- Substituir *TimestreamExecutionRole* com o nome da função que você está criando.
- Substituir *trust-policy* com o JSON arquivo que contém a política de confiança.

```
aws iam create-role --role-name TimestreamExecutionRole --assume-role-policy-document file://trust-policy.json
```

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "timestreamTrustPolicy",
 "Effect": "Allow",
 "Principal": {
 "Service": "iotfleetwise.amazonaws.com"
 },
 "Action": "sts:AssumeRole",
 "Condition": {
 "StringEquals": {
 "aws:SourceArn": [
 "arn:aws:iotfleetwise:region:account-id:campaign/campaign-name"
]
 }
 }
 }
]
}
```

```

 "aws:SourceAccount": [
 "account-id"
]
 }
}
]
}

```

Crie uma política de permissões para dar FleetWise permissões à AWS IoT para gravar dados no Timestream. Para criar uma política de permissões, execute o seguinte comando.

Para criar uma política de permissões

- Substituir *AWSIoT Fleetwise Access Timestream Permissions Policy* com o nome da política que você está criando.
- Substituir *permissions-policy* com o nome do JSON arquivo que contém a política de permissões.

```
aws iam create-policy --policy-name AWSIoT Fleetwise Access Timestream Permissions Policy --policy-document file://permissions-policy.json
```

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "timestreamIngestion",
 "Effect": "Allow",
 "Action": [
 "timestream:WriteRecords",
 "timestream:Select",
 "timestream:DescribeTable"
],
 "Resource": "table-arn"
 },
 {
 "Sid": "timestreamDescribeEndpoint",
 "Effect": "Allow",
 "Action": [
 "timestream:DescribeEndpoints"
]
 }
]
}

```

```
],
 "Resource": "*"
 }
]
}
```

Para anexar a política de permissões à sua IAM função

1. Na saída, copie o Amazon Resource Name (ARN) da política de permissões.
2. Para anexar a política de IAM permissões à sua IAM função, execute o comando a seguir.
  - Substituir *permissions-policy-arn* com o ARN que você copiou na etapa anterior.
  - Substituir *TimestreamExecutionRole* com o nome da IAM função que você criou.

```
aws iam attach-role-policy --policy-arn permissions-policy-arn --role-
name TimestreamExecutionRole
```

Para obter mais informações, consulte [Gerenciamento de acesso para AWS recursos](#) no Guia IAM do usuário.

## Identity and Access Management para AWS IoT FleetWise

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. IAMos administradores controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) para usar os recursos de IoT AWS . FleetWise IAMé um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

### Tópicos

- [Público](#)
- [Autenticando com identidades](#)
- [Gerenciando acesso usando políticas](#)
- [Como a AWS IoT FleetWise funciona com IAM](#)
- [Exemplos de políticas baseadas em identidade para IoT AWS FleetWise](#)



- [Solução de problemas AWS de FleetWise identidade e acesso à IoT](#)

## Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz na AWS IoT FleetWise.

**Usuário do serviço** — Se você usa o FleetWise serviço de AWS IoT para fazer seu trabalho, seu administrador fornecerá as credenciais e as permissões de que você precisa. À medida que você usa mais FleetWise recursos de AWS IoT para fazer seu trabalho, talvez precise de permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se você não conseguir acessar um recurso na AWS IoT FleetWise, consulte. [Solução de problemas AWS de FleetWise identidade e acesso à IoT](#)

**Administrador de serviços** — Se você é responsável pelos FleetWise recursos de AWS IoT em sua empresa, provavelmente tem acesso total à IoT AWS . FleetWise É seu trabalho determinar quais FleetWise recursos e recursos de AWS IoT seus usuários do serviço devem acessar. Em seguida, você deve enviar solicitações ao IAM administrador para alterar as permissões dos usuários do serviço. Revise as informações nesta página para entender os conceitos básicos do IAM. Para saber mais sobre como sua empresa pode usar IAM a AWS IoT FleetWise, consulte. [Como a AWS IoT FleetWise funciona com IAM](#)

**IAM administrador** — Se você for IAM administrador, talvez queira saber detalhes sobre como criar políticas para gerenciar o acesso à AWS IoT FleetWise. Para ver exemplos de políticas FleetWise baseadas em identidade de AWS IoT que você pode usar em, consulte. IAM [Exemplos de políticas baseadas em identidade para IoT AWS FleetWise](#)

## Autenticando com identidades

A autenticação é como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como IAM usuário ou assumindo uma IAM função. Usuário raiz da conta da AWS

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Os usuários (do IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você entra como uma identidade federada, seu administrador configurou previamente a federação de identidades usando IAM funções. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login em AWS, consulte [Como fazer login Conta da AWS](#) no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para você mesmo assinar solicitações, consulte [Assinar AWS API solicitações](#) no Guia IAM do usuário.

Independente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte [Autenticação multifator](#) no Guia AWS IAM Identity Center do usuário e [Uso da autenticação multifator \(MFA\) AWS no Guia do IAMusuário](#).

## Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário raiz para tarefas diárias. Proteja as credenciais do usuário raiz e use-as para executar as tarefas que somente ele puder executar. Para ver a lista completa de tarefas que exigem que você faça login como usuário raiz, consulte [Tarefas que exigem credenciais de usuário raiz](#) no Guia do IAM usuário.

## Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas acessam Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos usar o AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um

conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter informações sobre o IAM Identity Center, consulte [O que é o IAM Identity Center?](#) no Guia do AWS IAM Identity Center usuário.

## Grupos e usuários do IAM

Um [IAMusuário](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos confiar em credenciais temporárias em vez de criar IAM usuários que tenham credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com IAM os usuários, recomendamos que você alterne as chaves de acesso. Para obter mais informações, consulte [Altere as chaves de acesso regularmente para casos de uso que exigem credenciais de longo prazo](#) no Guia do IAMusuário.

Um [IAMgrupo](#) é uma identidade que especifica uma coleção de IAM usuários. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdminse conceder a esse grupo permissões para administrar IAM recursos.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte [Quando criar um IAM usuário \(em vez de uma função\)](#) no Guia do IAM usuário.

## IAMfunções

Uma [IAMfunção](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas. É semelhante a um IAM usuário, mas não está associado a uma pessoa específica. Você pode assumir temporariamente uma IAM função no AWS Management Console [trocando de funções](#). Você pode assumir uma função chamando uma AWS API operação AWS CLI or ou usando uma personalizadaURL. Para obter mais informações sobre métodos de uso de funções, consulte [Métodos para assumir uma função](#) no Guia IAM do usuário.

IAMfunções com credenciais temporárias são úteis nas seguintes situações:

- **Acesso de usuário federado:** para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter

informações sobre funções para federação, consulte [Criação de uma função para um provedor de identidade terceirizado](#) no Guia IAM do usuário. Se você usa o IAM Identity Center, configura um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a uma função em IAM. Para obter informações sobre conjuntos de permissões, consulte [Conjuntos de Permissões](#) no Manual do Usuário do AWS IAM Identity Center .

- Permissões temporárias IAM de IAM usuário — Um usuário ou função pode assumir uma IAM função para assumir temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas — Você pode usar uma IAM função para permitir que alguém (um diretor confiável) em uma conta diferente acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre funções e políticas baseadas em recursos para acesso entre contas, consulte [Acesso a recursos entre contas IAM no Guia](#) do IAM usuário.
- Acesso entre serviços — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos na Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões do principal de chamada, usando um perfil de serviço ou um perfil vinculado a serviço.
- Sessões de acesso direto (FAS) — Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. FAS solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte [Encaminhar sessões de acesso](#).
- Função de serviço — Uma função de serviço é uma [IAM função](#) que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamente IAM. Para obter mais informações, consulte [Criação de uma função para delegar permissões a uma AWS service \(Serviço da AWS\)](#) no Guia do IAM usuário.
- Função vinculada ao serviço — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS) O serviço pode presumir a função de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS

e são de propriedade do serviço. Um IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.

- Aplicativos em execução na Amazon EC2 — Você pode usar uma IAM função para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo AWS CLI AWS API solicitações. Isso é preferível ao armazenamento de chaves de acesso na EC2 instância. Para atribuir uma AWS função a uma EC2 instância e disponibilizá-la para todos os aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que os programas em execução na EC2 instância recebam credenciais temporárias. Para obter mais informações, consulte [Como usar uma IAM função para conceder permissões a aplicativos executados em EC2 instâncias da Amazon](#) no Guia IAM do usuário.

Para saber se usar IAM funções ou IAM usuários, consulte [Quando criar uma IAM função \(em vez de um usuário\)](#) no Guia do IAM usuário.

## Gerenciando acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como JSON documentos. Para obter mais informações sobre a estrutura e o conteúdo dos documentos de JSON política, consulte [Visão geral das JSON políticas](#) no Guia IAM do usuário.

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e funções não têm permissões. Para conceder permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

IAMas políticas definem permissões para uma ação, independentemente do método usado para realizar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função do AWS Management Console AWS CLI, do ou do AWS API.

## Políticas baseadas em identidade

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Criação de IAM políticas no Guia](#) do IAMusuário.

As políticas baseadas em identidade podem ser categorizadas ainda adicionalmente como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte [Escolha entre políticas gerenciadas e políticas em linha no Guia](#) do IAMusuário.

## Políticas baseadas no recurso

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas de uma política baseada IAM em recursos.

## Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento JSON de política.

Amazon S3, AWS WAF, e Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a [visão geral da lista de controle de acesso \(ACL\)](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

## Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- **Limites de permissões** — Um limite de permissões é um recurso avançado no qual você define as permissões máximas que uma política baseada em identidade pode conceder a uma IAM entidade (IAM usuário ou função). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte [Limites de permissões para IAM entidades](#) no Guia IAM do usuário.
- **Políticas de controle de serviço (SCPs)** — SCPs são JSON políticas que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em AWS Organizations. AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. Os SCP limites de permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations e SCPs, consulte [Políticas de controle de serviços](#) no Guia AWS Organizations do Usuário.
- **Políticas de sessão:** são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em atributo. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte [Políticas de sessão](#) no Guia IAM do usuário.

## Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de política estão envolvidos, consulte [Lógica de avaliação](#) de políticas no Guia IAM do usuário.

## Como a AWS IoT FleetWise funciona com IAM

Antes de usar IAM para gerenciar o acesso à AWS IoT FleetWise, saiba quais IAM recursos estão disponíveis para uso com a IoT AWS . FleetWise

IAMrecursos que você pode usar com a AWS IoT FleetWise

| IAMrecurso                                       | AWS Suporte à IoT FleetWise |
|--------------------------------------------------|-----------------------------|
| <a href="#">Políticas baseadas em identidade</a> | Sim                         |
| <a href="#">Políticas baseadas em recursos</a>   | Não                         |
| <a href="#">Ações das políticas</a>              | Sim                         |
| <a href="#">Atributos de políticas</a>           | Sim                         |
| <a href="#">Chaves de condição de políticas</a>  | Sim                         |
| <a href="#">ACLs</a>                             | Não                         |
| <a href="#">ABAC(tags nas políticas)</a>         | Parcial                     |
| <a href="#">Credenciais temporárias</a>          | Sim                         |
| <a href="#">Permissões de entidade principal</a> | Sim                         |
| <a href="#">Perfis de serviço</a>                | Não                         |
| <a href="#">Funções vinculadas ao serviço</a>    | Não                         |

Para ter uma visão geral de como a AWS FleetWise IoT e AWS outros serviços funcionam com a IAM maioria dos recursos, [AWS consulte os serviços que funcionam IAM](#) com no Guia IAMdo usuário.

### Políticas baseadas em identidade para IoT AWS FleetWise

Compatível com políticas baseadas em identidade: Sim



Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Criação de IAM políticas no Guia](#) do IAMusuário.

Com políticas IAM baseadas em identidade, você pode especificar ações e recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que você pode usar em uma JSON política, consulte a [referência IAM JSON de elementos de política](#) no Guia IAM do usuário.

## Exemplos de políticas baseadas em identidade para IoT AWS FleetWise

Para ver exemplos de políticas FleetWise baseadas em identidade de AWS IoT, consulte. [Exemplos de políticas baseadas em identidade para IoT AWS FleetWise](#)

## Políticas baseadas em recursos na IoT AWS FleetWise

Suporte a políticas baseadas em recursos: não

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para habilitar o acesso entre contas, você pode especificar uma conta ou IAM entidades inteiras em outra conta como principal em uma política baseada em recursos. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um IAM administrador na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a

uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, [consulte Acesso a recursos entre contas IAM no Guia do IAM usuário](#).

## Ações políticas para AWS IoT FleetWise

Compatível com ações de políticas: Sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O `Action` elemento de uma JSON política descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da AWS API operação associada. Há algumas exceções, como ações somente de permissão que não têm uma operação correspondente. API Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de FleetWise ações de AWS IoT, consulte [Ações definidas pela AWS IoT FleetWise](#) na Referência de autorização de serviço.

As ações políticas na AWS IoT FleetWise usam o seguinte prefixo antes da ação:

```
iotfleetwise
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

```
"Action": [
 "iotfleetwise:action1",
 "iotfleetwise:action2"
]
```

Você também pode especificar várias ações usando caracteres-curinga (\*). Por exemplo, para especificar todas as ações que começam com a palavra `List`, inclua a seguinte ação:

```
"Action": "iotfleetwise:List*"
```

Para ver exemplos de políticas FleetWise baseadas em identidade de AWS IoT, consulte [Exemplos de políticas baseadas em identidade para IoT AWS FleetWise](#)

## Recursos de políticas para AWS IoT FleetWise

Compatível com recursos de políticas: Sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Resource` JSON de política especifica o objeto ou objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu [Amazon Resource Name \(ARN\)](#). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (\*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Para ver uma lista dos tipos de FleetWise recursos de AWS IoT e seus ARNs, consulte [Recursos definidos pela AWS IoT FleetWise](#) na Referência de autorização de serviço. Para saber com quais ações você pode especificar cada recurso, consulte [Ações definidas pela AWS IoT FleetWise](#). ARN

Para ver exemplos de políticas FleetWise baseadas em identidade de AWS IoT, consulte [Exemplos de políticas baseadas em identidade para IoT AWS FleetWise](#)

## Chaves de condição de política para AWS IoT FleetWise

Compatível com chaves de condição de política específicas de serviço: Sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usem [agentes de condição](#), como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica AND. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma OR operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, você pode conceder permissão a um IAM usuário para acessar um recurso somente se ele estiver marcado com o nome de IAM usuário. Para obter mais informações, consulte [elementos de IAM política: variáveis e tags](#) no Guia IAM do usuário.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as [chaves de contexto de condição AWS global](#) no Guia IAM do usuário.

Para ver uma lista de chaves de FleetWise condição de AWS IoT, consulte [Chaves de condição para AWS IoT FleetWise](#) na Referência de autorização de serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte [Ações definidas pela AWS IoT FleetWise](#).

Para ver exemplos de políticas FleetWise baseadas em identidade de AWS IoT, consulte [Exemplos de políticas baseadas em identidade para IoT AWS FleetWise](#).

## Listas de controle de acesso (ACLs) na AWS IoT FleetWise

SuportesACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento JSON de política.

## Controle de acesso baseado em atributos (ABAC) com IoT AWS FleetWise

Suportes ABAC (tags nas políticas): Parciais

O controle de acesso baseado em atributos (ABAC) é uma estratégia de autorização que define permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a IAM entidades (usuários ou funções) e a muitos AWS recursos. Marcar entidades e recursos é a primeira etapa do ABAC. Em seguida, você cria ABAC políticas para permitir operações quando a tag do diretor corresponde à tag do recurso que ele está tentando acessar.

ABACé útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna complicado.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no [elemento de condição](#) de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobreABAC, consulte [O que éABAC?](#) no Guia do IAM usuário. Para ver um tutorial com etapas de configuraçãoABAC, consulte [Usar controle de acesso baseado em atributos \(ABAC\) no Guia](#) do IAMusuário.

#### Note

AWS A IoT FleetWise só oferece suporte a `iam:PassRole`, o que é necessário para a `CreateCampaign` API operação.

## Usando credenciais temporárias com IoT AWS FleetWise

Compatível com credenciais temporárias: Sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte [Serviços da AWS esse trabalho IAM](#) no Guia do IAM usuário.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre a troca de funções, consulte [Alternando para uma função \(console\)](#) no Guia IAM do usuário.

Você pode criar manualmente credenciais temporárias usando o AWS CLI ou AWS API. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte [Credenciais de segurança temporárias emIAM](#).

## Permissões principais entre serviços para IoT AWS FleetWise

Suporta sessões de acesso direto (FAS): Sim

Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. FAS as solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte [Encaminhar sessões de acesso](#).

## Funções de serviço para AWS IoT FleetWise

Compatível com perfis de serviço: não

Uma função de serviço é uma [IAM função](#) que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamente IAM. Para obter mais informações, consulte [Criação de uma função para delegar permissões a uma AWS service \(Serviço da AWS\)](#) no Guia do IAM usuário.

### Warning

Alterar as permissões de uma função de serviço pode interromper a funcionalidade AWS da IoT FleetWise. Edite funções de serviço somente quando a AWS IoT FleetWise fornecer orientação para fazer isso.

## Funções vinculadas a serviços para IoT AWS FleetWise

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode presumir a função de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.

Para obter detalhes sobre como criar ou gerenciar funções vinculadas a serviços, consulte [AWS serviços que funcionam](#) com IAM. Encontre um serviço na tabela que inclua um Yes na coluna Função vinculada ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a esse serviço.

## Usando funções vinculadas a serviços para IoT AWS FleetWise

AWS IoT FleetWise usa AWS Identity and Access Management (IAM) funções vinculadas ao [serviço](#). Uma função vinculada a serviços é um tipo exclusivo de IAM função vinculada diretamente à IoT AWS. As funções vinculadas ao serviço são predefinidas pela AWS IoT FleetWise e incluem as permissões que a IoT AWS precisa para enviar métricas para a Amazon FleetWise . CloudWatch Para obter mais informações, consulte [Monitore AWS a IoT com a Amazon FleetWise CloudWatch](#).

Uma função vinculada ao serviço agiliza a configuração da AWS FleetWise IoT porque você não precisa adicionar manualmente as permissões necessárias. AWS IoT FleetWise define as permissões de suas funções vinculadas a serviços e, a menos que seja definido de outra forma, somente a AWS FleetWise IoT pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões. Essa política de permissões não pode ser anexada a nenhuma outra IAM entidade.

Uma função vinculada ao serviço poderá ser excluída somente após excluir seus recursos relacionados. Isso protege seus FleetWise recursos de AWS IoT porque você não pode remover inadvertidamente a permissão para acessar os recursos.

Para obter informações sobre outros serviços que oferecem suporte a funções vinculadas a serviços, consulte [AWS serviços que funcionam com IAM](#) e procure os serviços que têm Sim na coluna Funções vinculadas ao serviço. Para visualizar a documentação da função vinculada ao serviço desse serviço, escolha Sim com um link.

### Permissões de função vinculadas a serviços para IoT AWS FleetWise

AWS IoT FleetWise usa a função vinculada ao serviço chamada `AWSServiceRoleForIoT FleetWise`— Uma política AWS gerenciada que é usada para todas as permissões out-of-the-box da IoT. AWS FleetWise

A função `AWSServiceRoleForIoT FleetWise` vinculada ao serviço confia nos seguintes serviços para assumir a função:

- `IoTFleetWise`

A política de permissões de função nomeada `AWSIoT FleetwiseServiceRolePolicy` permite que FleetWise a AWS IoT conclua as seguintes ações nos recursos especificados:

- Ação: `cloudwatch:PutMetricData` no recurso \*

Você deve configurar permissões para permitir que uma IAM entidade (como um usuário, grupo ou função) crie, edite ou exclua uma função vinculada ao serviço. Para obter mais informações, consulte [Permissões de funções vinculadas ao serviço](#) no Guia do IAMusuário.

### Criação de uma função vinculada a serviços para IoT AWS FleetWise

Não é necessário criar manualmente uma função vinculada ao serviço. Quando você registra uma conta no FleetWise console de AWS IoT AWS CLI, a ou a AWS IoT FleetWise cria a AWS API função vinculada ao serviço para você. Para obter mais informações, consulte [Defina suas configurações AWS de IoT FleetWise](#).

### Criação de uma função vinculada a serviços na AWS IoT (console) FleetWise

Não é necessário criar manualmente uma função vinculada ao serviço. Quando você registra uma conta no FleetWise console de AWS IoT AWS CLI, a ou a AWS IoT FleetWise cria a AWS API função vinculada ao serviço para você.

### Editando uma função vinculada a serviços para IoT AWS FleetWise

Você não pode editar a função `AWSServiceRoleForIoT FleetWise` vinculada ao serviço na IoT AWS. FleetWise Como várias entidades podem fazer referência a qualquer perfil vinculado ao serviço criado, não é possível alterar o nome do perfil. No entanto, você pode editar a descrição da função usando IAM. Para obter mais informações, consulte [Edição de uma função vinculada ao serviço](#) no Guia do IAMusuário.

### Limpar um perfil vinculado ao serviço

Antes de poder usar IAM para excluir uma função vinculada ao serviço, você deve primeiro excluir todos os recursos usados pela função.

#### Note

Se a AWS IoT FleetWise estiver usando a função quando você tentar excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação



novamente. Para saber como excluir o service-linked-role por meio do console,, ou AWS CLI AWS API, consulte [Usando funções vinculadas ao serviço no Guia](#) do IAMusuário.

Se você excluir essa função vinculada ao serviço e precisar criá-la novamente, poderá registrar uma conta na IoTAWS. FleetWise AWS FleetWise Em seguida, a IoT cria a função vinculada ao serviço para você novamente.

## Exemplos de políticas baseadas em identidade para IoT AWS FleetWise

Por padrão, usuários e funções não têm permissão para criar ou modificar recursos de AWS IoT FleetWise. Eles também não podem realizar tarefas usando o AWS Management Console, AWS Command Line Interface (AWS CLI) ou AWS API. Para conceder permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

Para saber como criar uma política IAM baseada em identidade usando esses exemplos de documentos de JSON política, consulte [Criação de IAM políticas no Guia](#) do IAMusuário.

Para obter detalhes sobre ações e tipos de recursos definidos pela AWS IoT FleetWise, incluindo o formato do ARNs para cada um dos tipos de recursos, consulte [Ações, recursos e chaves de condição para AWS IoT FleetWise](#) na Referência de Autorização de Serviço.

### Tópicos

- [Melhores práticas de política](#)
- [Usando o console AWS de IoT FleetWise](#)
- [Permitir que usuários visualizem suas próprias permissões](#)
- [Acessar recursos no Amazon Timestream](#)

## Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir FleetWise recursos de AWS IoT em sua conta. Essas ações podem incorrer em custos para seus Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas

AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte [políticas AWS gerenciadas](#) ou [políticas AWS gerenciadas para funções de trabalho](#) no Guia IAM do usuário.

- Aplique permissões com privilégios mínimos — Ao definir permissões com IAM políticas, conceda somente as permissões necessárias para realizar uma tarefa. Você faz isso definindo as ações que podem ser executadas em atributos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre IAM como usar para aplicar permissões, consulte [Políticas e permissões IAM no](#) Guia IAM do usuário.
- Use condições nas IAM políticas para restringir ainda mais o acesso — Você pode adicionar uma condição às suas políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte [Elementos IAM JSON da política: Condição](#) no Guia IAM do usuário.
- Use o IAM Access Analyzer para validar suas IAM políticas e garantir permissões seguras e funcionais — o IAM Access Analyzer valida políticas novas e existentes para que as políticas sigam a linguagem da IAM política (JSON) e as melhores práticas. IAM IAMO Access Analyzer fornece mais de 100 verificações de políticas e recomendações práticas para ajudá-lo a criar políticas seguras e funcionais. Para obter mais informações, consulte [Validação da política do IAM Access Analyzer](#) no Guia do IAM Usuário.
- Exigir autenticação multifatorial (MFA) — Se você tiver um cenário que exija IAM usuários ou um usuário root Conta da AWS, ative MFA para obter segurança adicional. Para exigir MFA quando API as operações são chamadas, adicione MFA condições às suas políticas. Para obter mais informações, consulte [Configurando o API acesso MFA protegido](#) no Guia do IAM usuário.

Para obter mais informações sobre as melhores práticas em IAM, consulte [as melhores práticas de segurança IAM no](#) Guia IAM do usuário.

## Usando o console AWS de IoT FleetWise

Para acessar o FleetWise console de AWS IoT, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os FleetWise recursos de AWS IoT em seu. Conta da AWS Se você criar uma política baseada em identidade que seja mais

restritiva do que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para AWS CLI o. ou AWS API o. Em vez disso, permita o acesso somente às ações que correspondam à API operação que eles estão tentando realizar.

Para garantir que usuários e funções ainda possam usar o FleetWise console de AWS IoT, anexe também a AWS IoT FleetWise ConsoleAccess ou a política ReadOnly AWS gerenciada às entidades. Para obter mais informações, consulte [Adicionar permissões a um usuário](#) no Guia do IAM usuário.

## Permitir que usuários visualizem suas próprias permissões

Este exemplo mostra como você pode criar uma política que permita IAM aos usuários visualizar as políticas embutidas e gerenciadas que estão anexadas à identidade do usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando o AWS CLI ou. AWS API

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ViewOwnUserInfo",
 "Effect": "Allow",
 "Action": [
 "iam:GetUserPolicy",
 "iam:ListGroupsWithUser",
 "iam:ListAttachedUserPolicies",
 "iam:ListUserPolicies",
 "iam:GetUser"
],
 "Resource": ["arn:aws:iam::*:user/${aws:username}"]
 },
 {
 "Sid": "NavigateInConsole",
 "Effect": "Allow",
 "Action": [
 "iam:GetGroupPolicy",
 "iam:GetPolicyVersion",
 "iam:GetPolicy",
 "iam:ListAttachedGroupPolicies",

```

```
 "iam:ListGroupPolicies",
 "iam:ListPolicyVersions",
 "iam:ListPolicies",
 "iam:ListUsers"
],
 "Resource": "*"
}
]
```

## Acessar recursos no Amazon Timestream

Antes de usar a AWS IoT FleetWise, você deve registrar sua AWS conta e os recursos do Amazon Timestream para AWS conceder permissão à FleetWise IoT para enviar dados do veículo em seu nome. IAM Nuvem AWS Veja os requisitos de registro:

- Um banco de dados do Amazon Timestream.
- Uma tabela criada no banco de dados do Amazon Timestream especificado.
- Uma IAM função que permite que a AWS IoT envie dados FleetWise para o Amazon Timestream.

Para obter mais informações, incluindo procedimentos e exemplos de políticas, consulte [Definir configurações](#).

## Solução de problemas AWS de FleetWise identidade e acesso à IoT

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com AWS FleetWise IoT e IAM

### Tópicos

- [Não estou autorizado a realizar uma ação na AWS IoT FleetWise](#)
- [Não estou autorizado a realizar iam: PassRole](#)
- [Quero permitir que pessoas de fora da minha Conta da AWS acessem meus recursos de AWS IoT FleetWise](#)

## Não estou autorizado a realizar uma ação na AWS IoT FleetWise

Se isso AWS Management Console indicar que você não está autorizado a realizar uma ação, entre em contato com o administrador para obter ajuda. Caso seu administrador seja a pessoa que forneceu suas credenciais de início de sessão.

O exemplo de erro a seguir ocorre quando o mateojackson IAM usuário tenta usar o console para ver detalhes sobre um *myVehicle* recurso fictício, mas não tem as `iotfleetwise:GetVehicleStatus` permissões.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
iotfleetwise:GetVehicleStatus on resource: myVehicle
```

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir a ele o acesso ao recurso *myVehicle* usando a ação `iotfleetwise:GetVehicleStatus`.

## Não estou autorizado a realizar iam: PassRole

Se você receber um erro informando que não está autorizado a realizar a `iam:PassRole` ação, suas políticas devem ser atualizadas para permitir que você passe uma função para a AWS IoT FleetWise.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando um IAM usuário chamado marymajor tenta usar o console para realizar uma ação na AWS IoT FleetWise. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

## Quero permitir que pessoas de fora da minha Conta da AWS acessem meus recursos de AWS IoT FleetWise

Você pode criar um perfil que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se a AWS IoT é FleetWise compatível com esses recursos, consulte [Como a AWS IoT FleetWise funciona com IAM](#)
- Para saber como fornecer acesso aos seus recursos em todas as Contas da AWS que você possui, consulte [Fornecer acesso a um IAM usuário em outra Conta da AWS de sua propriedade](#) no Guia do IAM usuário.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte [Fornecer Contas da AWS acesso a terceiros](#) no Guia do IAM usuário.
- Para saber como fornecer acesso por meio da federação de identidades, consulte [Fornecendo acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do IAM usuário.
- Para saber a diferença entre usar funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a [recursos entre contas IAM no Guia](#) do IAM usuário.

## Validação de conformidade para AWS IoT FleetWise

### Note


AWS IoT FleetWise não está no escopo de nenhum programa de AWS conformidade.

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte [Serviços da AWS Escopo por Programa de Conformidade](#) [Serviços da AWS](#) e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de [AWS conformidade](#) [Programas AWS](#) de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte [Baixar relatórios em AWS Artifact](#) .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- [Guias de início rápido sobre segurança e conformidade](#) — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos AWS focados em segurança e conformidade.
- [Arquitetura para HIPAA segurança e conformidade na Amazon Web Services](#) — Este whitepaper descreve como as empresas podem usar AWS para criar HIPAA aplicativos qualificados.

 Note

Nem todos Serviços da AWS são HIPAA elegíveis. Para obter mais informações, consulte a [Referência de serviços HIPAA elegíveis](#).

- AWS Recursos de <https://aws.amazon.com/compliance/resources/> de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- [AWS Guias de conformidade do cliente](#) — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização ()). ISO
- [Avaliação de recursos com regras](#) no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- [AWS Security Hub](#) — Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os recursos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a [Referência de controles do Security Hub](#).
- [Amazon GuardDuty](#) — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, por exemplo PCIDSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.

- [AWS Audit Manager](#)— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

## Resiliência na IoT AWS FleetWise

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade. As regiões fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, que são conectadas com baixa latência, alta throughput e redes altamente redundantes. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data centers tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte [Infraestrutura AWS global](#).

### Note

Os dados processados pela AWS IoT são FleetWise armazenados em um banco de dados Amazon Timestream. O Timestream suporta backups em outras zonas ou regiões de AWS disponibilidade. No entanto, você pode escrever seu próprio aplicativo usando o Timestream SDK para consultar dados e salvá-los no destino de sua escolha.

Para obter mais informações sobre o Amazon Timestream, consulte o [Guia do desenvolvedor do Amazon Timestream](#).

## Segurança da infraestrutura na AWS IoT FleetWise

Como um serviço gerenciado, a AWS IoT FleetWise é protegida pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte [AWS Cloud Security](#). Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte [Proteção](#) de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa API chamadas AWS publicadas para acessar a AWS IoT FleetWise pela rede. Os clientes devem oferecer suporte para:



- Segurança da camada de transporte (TLS). Exigimos TLS 1,2 e recomendamos TLS 1,3.
- Suítes de criptografia com sigilo direto perfeito (), como (Ephemeral PFS Diffie-Hellman) ou DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando uma ID de chave de acesso e uma chave de acesso secreta associada a um IAM principal. Ou você pode usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Você pode chamar essas API operações de qualquer local de rede, mas a AWS IoT FleetWise oferece suporte a políticas de acesso baseadas em recursos, que podem incluir restrições com base no endereço IP de origem. Você também pode usar FleetWise políticas de AWS IoT para controlar o acesso de endpoints específicos da Amazon Virtual Private Cloud VPC (Amazon) ou específicos. VPCs Efetivamente, isso isola o acesso à rede a um determinado recurso de AWS FleetWise IoT apenas do VPC específico dentro AWS da rede.

## Tópicos

- [Conectando-se à AWS IoT FleetWise por meio de um endpoint de interface VPC](#)

## Conectando-se à AWS IoT FleetWise por meio de um endpoint de interface VPC

Você pode se conectar diretamente à AWS IoT FleetWise usando um [VPC endpoint de interface \(AWS PrivateLink\)](#) na sua Virtual Private Cloud (VPC), em vez de se conectar pela Internet. Quando você usa um VPC endpoint de interface, a comunicação entre você VPC e a AWS FleetWise IoT é conduzida inteiramente dentro da AWS rede. Cada VPC endpoint é representado por uma ou mais [interfaces de rede elástica](#) (ENIs) com endereços IP privados em suas VPC sub-redes.

O VPC endpoint da interface conecta você VPC diretamente à AWS FleetWise IoT sem um gateway de internetNAT, dispositivoVPN, AWS Direct Connect conexão ou conexão. As instâncias na sua VPC não precisam de endereços IP públicos para se comunicar com a AWS IoT FleetWiseAPI.

Para usar a AWS IoT FleetWise por meio do seuVPC, você deve se conectar a partir de uma instância que esteja dentro da VPC ou conectar sua rede privada à sua VPC usando um AWS Virtual Private Network (VPN) ou. AWS Direct Connect Para obter informações sobre a AmazonVPN, consulte [VPNs conexões](#) no Guia do usuário da Amazon Virtual Private Cloud. Para obter

informações sobre AWS Direct Connect, consulte [Criação de uma conexão](#) no Guia AWS Direct Connect do usuário.

Você pode criar um VPC endpoint de interface para se conectar à AWS FleetWise IoT usando AWS o console AWS Command Line Interface ou os comandos AWS CLI(). Para obter mais informações, consulte [Creating an interface endpoint](#) (Criação de um endpoint de interface).

Depois de criar um VPC endpoint de interface, se você habilitar DNS nomes de host privados para o endpoint, o endpoint de AWS IoT padrão será resolvido para seu FleetWise endpoint. VPC O endpoint de nome de serviço padrão para AWS FleetWise IoT está no formato a seguir.

```
iotfleetwise.Region.amazonaws.com
```

Se você não habilitar DNS nomes de host privados, a Amazon VPC fornecerá um nome de DNS endpoint que você pode usar no seguinte formato.

```
VPCE_ID.iotfleetwise.Region.vpce.amazonaws.com
```

Para obter mais informações, consulte [VPCEndpoints de interface \(AWS PrivateLink\)](#) no Guia do VPC usuário da Amazon.

AWS A IoT FleetWise suporta a realização de chamadas para todas as suas [APIações](#) dentro do seu VPC

Você pode anexar políticas de VPC endpoint a um VPC endpoint para controlar o acesso dos principais. IAM Você também pode associar grupos de segurança a um VPC endpoint para controlar o acesso de entrada e saída com base na origem e no destino do tráfego de rede, como uma variedade de endereços IP. Para obter mais informações, consulte [Controle do acesso a serviços com VPC endpoints](#).

## Criação de uma política VPC de endpoint para IoT AWS FleetWise

Você pode criar uma política para VPC endpoints da Amazon para AWS FleetWise IoT para especificar o seguinte:

- O principal que pode ou não executar ações
- As ações que podem ou não ser executadas

Para obter mais informações, consulte [Controle do acesso a serviços com VPC endpoints](#) no Guia do VPC usuário da Amazon.

Example — política de VPC endpoint para negar todo o acesso de uma conta especificada AWS

A seguinte política de VPC endpoint nega AWS a conta **123456789012** todas as API chamadas usando o endpoint.

```
{
 "Statement": [
 {
 "Action": "*",
 "Effect": "Allow",
 "Resource": "*",
 "Principal": "*"
 },
 {
 "Action": "*",
 "Effect": "Deny",
 "Resource": "*",
 "Principal": {
 "AWS": [
 "123456789012"
]
 }
 }
]
}
```

Example — política de VPC endpoint para permitir VPC acesso somente a um IAM principal especificado (usuário)

A política de VPC endpoint a seguir permite acesso total somente ao usuário a **Lijuan** na AWS conta **123456789012**. Ele nega a todos os outros IAM diretores o acesso ao endpoint.

```
{
 "Statement": [
 {
 "Action": "*",
 "Effect": "Allow",
 "Resource": "*",
 "Principal": {
 "AWS": [
```

```
 "arn:aws:iam::123456789012:user/Lijuan"
]
}
}]
}
```

Example — política VPC de endpoint para ações de AWS IoT FleetWise

Veja a seguir um exemplo de uma política de endpoint para AWS IoT FleetWise. Quando anexada a um endpoint, essa política concede acesso às ações de AWS FleetWise IoT listadas para o usuário. IAM *fleetWise* no Conta da AWS *123456789012*.

```
{
 "Statement": [
 {
 "Principal": {
 "AWS": [
 "arn:aws:iam::123456789012:user/fleetWise"
],
 },
 "Resource": "*",
 "Effect": "Allow",
 "Action": [
 "iotfleetwise:ListFleets",
 "iotfleetwise:ListCampaigns",
 "iotfleetwise:CreateVehicle",
]
 }
]
}
```

## Análise de configuração e vulnerabilidade na AWS IoT FleetWise

Ambientes de IoT consistem em grandes quantidades de dispositivos com capacidades diversas, duradouros e geograficamente distribuídos. Essas características tornam a configuração do dispositivo complexa e propensa a erros. Além disso, os dispositivos quase sempre têm restrições quanto à capacidade computacional, memória e capacidade de armazenamento, o uso de criptografia e outras formas de segurança nos dispositivos é limitado. Os dispositivos geralmente usam software com vulnerabilidades conhecidas. Esses fatores tornam os dispositivos de IoT, incluindo veículos que coletam dados AWS para IoT FleetWise, um alvo atraente para hackers e dificultam sua proteção contínua.

A configuração e os controles de TI são uma responsabilidade compartilhada entre você AWS e você, nosso cliente. Para obter mais informações, consulte o [modelo de responsabilidade AWS compartilhada](#).

## Melhores práticas de segurança para AWS IoT FleetWise

AWS A IoT FleetWise fornece vários recursos de segurança a serem considerados ao desenvolver e implementar suas próprias políticas de segurança. As melhores práticas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas melhores práticas podem não ser adequadas ou suficientes para o seu ambiente, trate-as como considerações úteis em vez de prescrições.

Para saber mais sobre segurança em, AWS IoT consulte [Práticas recomendadas de segurança AWS IoT Core no](#) Guia do AWS IoT desenvolvedor.

### Conceder o mínimo possível de permissões

Siga o princípio do menor privilégio usando o conjunto mínimo de permissões nas IAM funções. Limite o uso do \* caractere curinga para as Resource propriedades Action e em suas IAM políticas. Em vez disso, declare um conjunto finito de ações e recursos quando possível. Para obter mais informações sobre as melhores práticas de privilégio mínimo e outras de políticas, consulte [the section called “Melhores práticas de política”](#).

### Não registrar em log informações confidenciais

Você deve evitar o registro de credenciais e outras informações de identificação pessoal (). PII Recomendamos que você implemente as seguintes proteções:

- Não use informações confidenciais nos nomes dos dispositivos.
- Não use informações confidenciais nos nomes e nos FleetWise recursos IDs de AWS IoT, por exemplo, em nomes de campanhas, manifestos de decodificadores, modelos de veículos e catálogos de sinais ou de veículos e frotas. IDs

### Use AWS CloudTrail para ver o histórico API de chamadas

Você pode ver um histórico das FleetWise API chamadas de AWS IoT feitas em sua conta para fins de análise de segurança e solução de problemas operacionais. Para receber um histórico

das FleetWise API chamadas de AWS IoT feitas em sua conta, basta ativar CloudTrail o. AWS Management Console Para obter mais informações, consulte [the section called “CloudTrail troncos”](#).

## Manter o relógio do dispositivo sincronizado

É importante ter a hora exata no seu dispositivo. Os certificados X.509 têm data e hora de expiração. O relógio em seu dispositivo é usado para verificar se um certificado de servidor ainda é válido. Os relógios do dispositivo podem atrasar ao longo do tempo ou as baterias podem descarregar.

Para obter mais informações, consulte a melhor prática [Manter o relógio do dispositivo sincronizado](#) no Guia do desenvolvedor do AWS IoT Core .

## Monitore a AWS IoT FleetWise

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho da AWS IoT FleetWise e de suas outras AWS soluções. AWS fornece as seguintes ferramentas de monitoramento para monitorar a AWS IoT FleetWise, relatar quando algo está errado e realizar ações automáticas quando apropriado:

- A Amazon CloudWatch monitora seus AWS recursos e os aplicativos nos quais você executa AWS em tempo real. É possível coletar e rastrear métricas, criar painéis personalizados e definir alarmes que notificam você ou que realizam ações quando uma métrica atinge um limite especificado. Por exemplo, você pode CloudWatch rastrear o CPU uso ou outras métricas de suas EC2 instâncias da Amazon e iniciar automaticamente novas instâncias quando necessário. Para obter mais informações, consulte o [Guia CloudWatch do usuário da Amazon](#).
- O Amazon CloudWatch Logs pode ser usado para monitorar, armazenar e acessar seus arquivos de log de EC2 instâncias da Amazon e de outras fontes. CloudTrail CloudWatch Os registros podem monitorar as informações nos arquivos de log e notificá-lo quando determinados limites forem atingidos. É possível também arquivar seus dados de log em armazenamento resiliente. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch Logs](#).
- AWS CloudTrail captura API chamadas e eventos relacionados feitos por ou em nome de você Conta da AWS. Em seguida, ele disponibiliza os arquivos de log a um bucket do Amazon S3 especificado por você. Você pode identificar quais usuários e contas ligaram AWS, o endereço IP de origem a partir do qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações, consulte o [Guia do usuário do AWS CloudTrail](#).

## Monitore AWS a IoT com a Amazon FleetWise CloudWatch

As CloudWatch métricas da Amazon são uma forma de monitorar seus AWS recursos e o desempenho deles. AWS A IoT FleetWise envia métricas para. CloudWatch Você pode usar o AWS Management Console AWS CLI, o ou an API para listar as métricas para as quais a AWS IoT FleetWise envia. CloudWatch Para obter mais informações, consulte o [Guia CloudWatch do usuário da Amazon](#).

**⚠ Important**

Você deve definir as configurações para que a AWS IoT FleetWise possa enviar métricas para o CloudWatch. Para obter mais informações, consulte [Defina suas configurações AWS de IoT FleetWise](#).

O namespace `AWS/IoTFleetWise` inclui as métricas a seguir.

**Métricas de sinal**

| Métrica                             | Descrição                                                                                                                                                                                                                                                                                 |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>IllegalMessageFromEdge</code> | <p>Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise não correspondia ao formato exigido.</p> <p>Unidades: contagem</p> <p>Dimensões: <code>VehicleName</code></p> <p>Estatística válida: soma</p>                                                                        |
| <code>MessageThrottled</code>       | <p>Uma mensagem enviada do veículo para a AWS IoT FleetWise foi estrangulada. Isso ocorre porque você excedeu os <a href="#">limites de serviço</a> dessa conta na região atual.</p> <p>Unidades: contagem</p> <p>Dimensões: <code>VehicleName</code></p> <p>Estatística válida: soma</p> |
| <code>ModelingError</code>          | <p>Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise contém sinais que não são validados em relação ao modelo do veículo.</p> <p>Unidades: contagem</p>                                                                                                                   |



| Métrica       | Descrição                                                                                                                                                                                                                                            |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <p>Dimensões: ModelManifestName</p> <p>Estatística válida: soma</p>                                                                                                                                                                                  |
| DecodingError | <p>Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise contém sinais que falham na decodificação do manifesto do decodificador do veículo.</p> <p>Unidades: contagem</p> <p>Dimensões: DecoderName</p> <p>Estatística válida: soma</p> |

### Métricas de campanha

| Métrica         | Descrição                                                                                                                                                                                          |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VehicleNotFound | <p>Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise, em que o veículo é desconhecido.</p> <p>Unidades: contagem</p> <p>Dimensões: VehicleName</p> <p>Estatística válida: soma</p> |
| CampaignInvalid | <p>Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise, em que a campanha não é válida.</p> <p>Unidades: contagem</p> <p>Dimensões: CampaignName</p> <p>Estatística válida: soma</p> |

| Métrica          | Descrição                                                                                                                                                                                          |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CampaignNotFound | <p>Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise, onde a campanha é desconhecida.</p> <p>Unidades: contagem</p> <p>Dimensões: CampaignName</p> <p>Estatística válida: soma</p> |

### Métricas de destino dos dados da campanha

| Métrica              | Descrição                                                                                                                                                                                                              |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TimestreamWriteError | <p>AWS A IoT não FleetWise conseguiu gravar uma mensagem do veículo na tabela Amazon Timestream.</p> <p>Unidades: contagem</p> <p>Dimensões: DatabaseName, TableName</p> <p>Estatística válida: soma</p>               |
| S3 WriteError        | <p>AWS A IoT não FleetWise conseguiu gravar uma mensagem do veículo no bucket do Amazon Simple Storage Service (Amazon S3).</p> <p>Unidades: contagem</p> <p>Dimensões: BucketName</p> <p>Estatística válida: soma</p> |
| S3 ReadError         | <p>AWS A IoT não FleetWise conseguiu ler uma chave de objeto do veículo no bucket do Amazon Simple Storage Service (Amazon S3).</p>                                                                                    |

| Métrica | Descrição                |
|---------|--------------------------|
|         | Unidades: contagem       |
|         | Dimensões: BucketName    |
|         | Estatística válida: soma |

### AWS KMS Principais métricas gerenciadas pelo cliente

| Métrica            | Descrição                                                                                                                                                                                                                                                      |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| KMSKeyAccessDenied | <p>AWS A IoT não FleetWise conseguiu gravar uma mensagem do veículo na tabela Timestream ou no bucket do Amazon S3 devido a um erro de acesso negado à chave. AWS KMS</p> <p>Unidades: contagem</p> <p>Dimensões: KMSKeyId</p> <p>Estatística válida: soma</p> |

## Monitore AWS a IoT com o FleetWise Amazon Logs CloudWatch

O Amazon CloudWatch Logs monitora os eventos que ocorrem em seus recursos e alerta você se houver algum problema. Se receber um alerta, você poderá acessar os arquivos de log para obter informações sobre o evento específico. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch Logs](#).

### Exibir FleetWise registros AWS de IoT no console CloudWatch

#### Important

Antes de ver o grupo de FleetWise registros de AWS IoT no CloudWatch console, verifique se o seguinte é verdadeiro:

- Você ativou o login na AWS IoT FleetWise. Para obter mais informações sobre registro em log, consulte [Configurar o AWS registro de IoT FleetWise](#).

- Já existem entradas de registro gravadas por AWS IoT operações.

Para ver seus FleetWise registros de AWS IoT no console CloudWatch

1. Abra o [CloudWatch console](#).
2. No painel de navegação, escolha Logs > Grupos de logs.
3. Escolha o grupo de logs .
4. Escolha Pesquisar grupo de logs. Você verá uma lista completa dos eventos de log gerados para sua conta.
5. Escolha o ícone de expansão para examinar um fluxo individual e encontrar todos os logs que tem um nível de log ERROR.

Você também pode inserir uma consulta na caixa de pesquisa Filtrar eventos. Por exemplo, você pode tentar a seguinte consulta:

```
{ $.logLevel = "ERROR" }
```

Para obter mais informações sobre a criação de expressões de filtro, consulte [Sintaxe de filtro e padrão](#) no Guia do usuário do Amazon CloudWatch Logs.

Example entrada de registro

```
{
 "accountId": "123456789012",
 "vehicleName": "test-vehicle",
 "message": "Unrecognized signal ID",
 "eventType": "MODELING_ERROR",
 "logLevel": "ERROR",
 "timestamp": 1685743214239,
 "campaignName": "test-campaign",
 "signalCatalogName": "test-catalog",
 "signalId": 10242
}
```

## Tipos de eventos de sinal

| Tipo de evento            | Descrição                                                                                                                                                                                                                                                                                        |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MODELING_ERROR            | <p>Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise contém sinais que não são validados em relação ao modelo do veículo.</p> <p>Atributos: vehicleName,campaignName, signalCatalogName,signalId,,signalValue, signalValueRange Min, signalValueRange Max, modelManifestName</p> |
| ILLEGAL_MESSAGE_FROM_EDGE | <p>Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise não correspondia ao formato exigido.</p> <p>Atributos:vehicleName,campaignName, signalCatalogName</p>                                                                                                                       |
| DECODING_ERROR            | <p>Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise contém sinais que falham na decodificação do manifesto do decodificador do veículo.</p> <p>Atributos: campaignName signalCatalogName, decoderManifestName,, (opcional)signalName, (opcional) s3 URI</p>                     |

## Tipos de evento de campanha

| Tipo de evento    | Descrição                                                                                                                                             |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| VEHICLE_NOT_FOUND | <p>Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise, onde o veículo era desconhecido.</p> <p>Atributos:vehicleName, campaignName</p> |

| Tipo de evento     | Descrição                                                                                                                                                          |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CAMPAIGN_NOT_FOUND | <p>Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise, onde a campanha era desconhecida.</p> <p>Atributos: vehicleName (opcional), campaignName</p> |
| CAMPAIGN_INVALID   | <p>Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise, em que a campanha não era válida.</p> <p>Atributos: vehicleName (opcional), campaignName</p> |

#### Tipos de eventos de destino de dados da campanha

| Tipo de evento         | Descrição                                                                                                                                                                         |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TIMESTREAM_WRITE_ERROR | <p>AWS IoT não conseguiu gravar uma mensagem do veículo na tabela Amazon Timestream.</p> <p>Atributos: vehicleName, campaignName, timestreamDatabaseName, timestreamTableName</p> |
| S3_WRITE_ERROR         | <p>AWS IoT não conseguiu gravar uma mensagem do veículo no bucket do Amazon Simple Storage Service (Amazon S3).</p> <p>Atributos: campaignName, destinationName</p>               |
| S3_READ_ERROR          | <p>AWS IoT não conseguiu ler uma chave de objeto do veículo no bucket do Amazon Simple Storage Service (Amazon S3).</p> <p>Atributos: campaignName, destinationName</p>           |

## Tipos de eventos AWS KMS principais gerenciados pelo cliente

| Tipo de evento        | Descrição                                                                                                                                                          |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| KMS_KEY_ACCESS_DENIED | AWS A IoT não FleetWise conseguiu gravar uma mensagem do veículo na tabela Timestream ou no bucket do Amazon S3 devido a um erro de acesso negado à chave. AWS KMS |

## Atributos

Todas as entradas de CloudWatch registros incluem os seguintes atributos:

`accountId`

Sua Conta da AWS identidade.

`eventType`

O tipo de evento para o qual o log foi gerado. O valor do tipo de evento depende do evento que gerou a entrada de log. Cada descrição de entrada de log inclui o valor de `eventType` para essa entrada de log.

`logLevel`

O nível de log que está sendo usado. Para obter mais informações, consulte [Níveis de log](#) no Guia do desenvolvedor do AWS IoT Core .

`message`

Contém detalhes específicos sobre o log.

`timestamp`

A data e hora de milissegundos da época em que a IoT AWS processou o registro. FleetWise

## Atributos opcionais

CloudWatch As entradas de registros incluem opcionalmente esses atributos, dependendo de `eventType`:

**decoderManifestName**

O nome do manifesto do decodificador que contém o sinal.

**destinationName**

O nome do destino dos dados do veículo. Por exemplo, o nome do bucket do Amazon S3.

**campaignName**

O nome da campanha.

**signalCatalogName**

O nome do catálogo de sinais que contém o sinal.

**signalId**

O ID do sinal de erro.

**signalIds**

Uma lista de sinais de erroIDs.

**signalName**

O nome do sinal.

**signalTimestampEpochSrta**

A marcação de data/hora do sinal de erro.

**signalValue**

O valor do sinal de erro.

**signalValueRangeMáximo**

O alcance máximo do sinal de erro.

**signalValueRangeMin**

O alcance mínimo do sinal de erro.

**s3 URI**

O identificador exclusivo do Amazon S3 de um arquivo do Amazon Ion de uma mensagem do veículo.



## timestreamDatabaseName

O nome do banco de dados do Timestream.

## timestreamTableName

O nome da tabela do Timestream.

## vehicleName

O nome do veículo.

## Configurar o AWS registro de IoT FleetWise

Você pode enviar seus dados de FleetWise registro de AWS IoT para um grupo de CloudWatch registros. CloudWatch Os registros oferecem visibilidade caso a AWS IoT FleetWise falhe em processar mensagens de veículos. Por exemplo, isso pode acontecer devido a uma configuração incorreta ou a outros erros do cliente. Você receberá notificações sobre quaisquer erros para poder identificar e mitigar problemas.

Antes de enviar registros para CloudWatch, você deve criar um grupo de CloudWatch registros. Configure o grupo de registros com a mesma conta e na mesma região que você usou com a AWS IoT FleetWise. Ao habilitar o login na AWS IoT FleetWise, forneça o nome do grupo de registros. Depois que o registro é ativado, a AWS IoT FleetWise entrega os registros ao grupo de CloudWatch registros em fluxos de registros.

Você pode visualizar os dados de log enviados da AWS IoT FleetWise no CloudWatch console. Para obter mais informações sobre como configurar um grupo de CloudWatch registros e visualizar dados de registros, consulte [Trabalhando com grupos de registros](#).

## Permissões para publicar registros em CloudWatch

A configuração do registro em CloudWatch log para um grupo de registros requer as configurações de permissões descritas nesta seção. Para obter informações sobre o gerenciamento de permissões, consulte [Gerenciamento de acesso para AWS recursos](#) no Guia IAM do usuário.

Com essas permissões, você pode alterar a configuração de registro, configurar a entrega de CloudWatch registros e recuperar informações sobre seu grupo de registros.

```
{
 "Version": "2012-10-17",
 "Statement": [
```

```
{
 "Action": [
 "iotfleetwise:PutLoggingOptions",
 "iotfleetwise:GetLoggingOptions"
],
 "Resource": [
 "*"
],
 "Effect": "Allow",
 "Sid": "IoTFleetwiseLoggingOptionsAPI"
}
{
 "Sid": "IoTFleetwiseLoggingCWL",
 "Action": [
 "logs:CreateLogDelivery",
 "logs:GetLogDelivery",
 "logs:UpdateLogDelivery",
 "logs>DeleteLogDelivery",
 "logs>ListLogDeliveries",
 "logs:PutResourcePolicy",
 "logs:DescribeResourcePolicies",
 "logs:DescribeLogGroups"
],
 "Resource": [
 "*"
],
 "Effect": "Allow"
}
]
```

Quando as ações são permitidas em todos os AWS recursos, isso é indicado na política com uma "Resource" configuração de "\*". Isso significa que as ações são permitidas em todos os AWS recursos que cada ação suporta.

## Configurar o registro em AWS IoT FleetWise (console)

Esta seção descreve como usar o FleetWise console de AWS IoT para configurar o registro.

Para usar o FleetWise console de AWS IoT para configurar o registro

1. Abra o console [AWS de IoT. FleetWise](#)

2. No painel esquerdo, escolha Settings (Configurações).
3. Na seção Registro em log da página Configurações, selecione Editar.
4. Na seção de CloudWatch registro, insira o grupo Registro.
5. Selecione Enviar para salvar as alterações.

Depois de habilitar o registro, você pode visualizar seus dados de registro no [CloudWatch console](#).

## Configurar o registro padrão no AWS IoT ( FleetWise ) CLI

Esta seção descreve como configurar o registro em log para AWS IoT FleetWise usando o CLI

Você também pode realizar esse procedimento com o API usando os métodos no AWS API que correspondem aos CLI comandos mostrados aqui. Você pode usar a [GetLoggingOptions](#) API operação para buscar a configuração atual e a [PutLoggingOptions](#) API operação para modificar a configuração.

Para usar o CLI para configurar o registro para AWS IoT FleetWise

1. Para obter as opções de registro em log da sua conta, use o comando get-logging-options.

```
aws iotfleetwise get-logging-options
```

2. Para ativar o registro em log, use o comando put-logging-options.

```
aws iotfleetwise put-logging-options --cloud-watch-log-delivery
logType=ERROR,logGroupName=MyLogGroup
```

onde:

logType

O tipo de registro para enviar dados para o CloudWatch Logs. Para desativar o registro em log, altere o valor para OFF.

logGroupName

O grupo de CloudWatch registros para o qual a operação envia dados. Certifique-se de criar o nome do grupo de registros antes de habilitar o registro para AWS IoT FleetWise.

Depois de ativar o registro, consulte [Pesquisar entradas do registro usando AWS CLI](#) o.

# Registre chamadas de AWS IoT usando FleetWise API AWS CloudTrail

AWS A IoT FleetWise é integrada com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço na IoT AWS . FleetWise CloudTrail captura todas as API chamadas para AWS FleetWise IoT como eventos. As chamadas capturadas incluem chamadas do FleetWise console de AWS IoT e chamadas de código para as operações de IoT AWS . FleetWise API Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para IoT AWS . FleetWise Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita à AWS IoT FleetWise, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre isso CloudTrail, consulte o [Guia AWS CloudTrail do usuário](#).

## AWS Informações de IoT em FleetWise CloudTrail

CloudTrail é ativado em sua AWS conta quando você cria a conta. Quando a atividade ocorre na AWS IoT FleetWise, essa atividade é registrada em um CloudTrail evento junto com outros eventos de AWS serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para obter mais informações, consulte [Visualização de eventos com histórico de CloudTrail eventos](#).

Para um registro contínuo dos eventos em sua AWS conta, incluindo eventos de AWS IoT FleetWise, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, a trilha se aplica a todas as AWS regiões. A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte as informações a seguir.

- [Visão geral da criação de uma trilha](#)
- [CloudTrail serviços e integrações suportados](#)
- [Configurando SNS notificações da Amazon para CloudTrail](#)
- [Recebendo arquivos de CloudTrail log de várias regiões](#)
- [Recebendo arquivos de CloudTrail log de várias contas](#)

[Todas as FleetWise ações de AWS IoT são registradas CloudTrail e documentadas na Referência de IoT AWS . FleetWise API](#) Por exemplo, chamadas para as `GetModelManifest` ações `CreateCampaignAssociateVehicleFleet`, e geram entradas nos arquivos de CloudTrail log.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais raiz ou de IAM usuário.
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte o [CloudTrail userIdentityelemento](#).

## Entenda as entradas do arquivo de FleetWise log AWS de IoT

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das API chamadas públicas, portanto, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a *AssociateVehicleFleet* operação.

```
{
 "eventVersion": "1.05",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "AIDACKCEVSQ6C2EXAMPLE",
 "arn": "arn:aws:iam::111122223333:assumed-role/NikkiWolf",
 "accountId": "111122223333",
 "accessKeyId": "access-key-id",
 "userName": "NikkiWolf"
 },
 "eventTime": "2021-11-30T09:56:35Z",
 "eventSource": "iotfleetwise.amazonaws.com",
 "eventName": "AssociateVehicleFleet",
 "awsRegion": "us-east-1",
```

```
"sourceIPAddress": "192.0.2.21",
"userAgent": "aws-cli/2.3.2 Python/3.8.8 Darwin/18.7.0 botocore/2.0.0",
"requestParameters": {
 "fleetId": "f1234567890",
 "vehicleId": "v0213456789"
},
"responseElements": {
},
"requestID": "9f861429-11e3-11e8-9eea-0781b5c0ac21",
"eventID": "17385819-4927-41ee-a6a5-29ml0br812v4",
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}
```

# Histórico de documentos do Guia do AWS desenvolvedor de IoT FleetWise

A tabela a seguir descreve as versões da documentação para AWS IoT FleetWise.

| Alteração                                               | Descrição                                                                                                                                                                                                                                                                                                                                                                                                 | Data                   |
|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <a href="#">Prévia dos dados de sistemas de visão</a>   | Você pode usar a visualização prévia dos dados do sistema de visão da AWS IoT FleetWise para coletar e organizar dados dos sistemas de visão do veículo, inclusive de câmeras, radares e lidars. Ele mantém os dados de sistemas de visão estruturados e não estruturados, os metadados (ID do evento, campanha, veículo) e o sensor padrão (dados de telemetria) sincronizados automaticamente na nuvem. | 26 de novembro de 2023 |
| <a href="#">AWS KMS chaves gerenciadas pelo cliente</a> | AWS A IoT FleetWise agora oferece suporte a chaves gerenciadas pelo AWS KMS cliente. Você pode usar a KMS chave para criptografar dados do lado do servidor relacionados aos FleetWise recursos de AWS IoT (catálogo de sinais, modelo do veículo, manifesto do decodificador, veículos e configurações da                                                                                                | 16 de outubro de 2023  |

campanha de coleta de dados)  
armazenados. Nuvem AWS

[Armazenamento de objetos no Amazon S3](#)

AWS A IoT FleetWise agora suporta o armazenamento de dados usando o Amazon Simple Storage Service (Amazon S3). É possível armazenar dados coletados durante campanhas no Amazon S3, além do Amazon Timestream.

1.º de junho de 2023

[Disponibilidade geral](#)

Este é o lançamento público da AWS IoT FleetWise.

27 de setembro de 2022

[Lançamento inicial](#)

Esta é a versão prévia do Guia do FleetWise Desenvolvedor de AWS IoT.

30 de novembro de 2021



As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.